

Table of Contents

Part 1: Neural networks and applications.....	1
Chapter 1: Handwriting knowledge based on parameterization for writer identification	3
<i>Carlos F. Romero, Carlos M. Travieso, Miguel A. Ferrer, Jesús B. Alonso</i>	
Chapter 2: Identification surfaces family.....	15
<i>Virgil Teodor, Alexandru Epureanu, Ciprian Cuzmin</i>	
Chapter 3: Neuro-fuzzy models and tobacco control	27
<i>Sonja Petrovic-Lazarevic, Jian Ying Zhang</i>	
Chapter 4: PNN for molecular level selection detection.....	35
<i>Krzysztof A. Cyran</i>	
Chapter 5: Fine-tune Artificial Neural Networks automatically	43
<i>Francisco Reinaldo, Rui Camacho, Luís P. Reis, Demétrio Renó Magalhães</i>	
Chapter 6: A neurofuzzy network for supporting detection of diabetic symptoms.....	49
<i>Leonarda Carnimeo</i>	
Chapter 7: Empirical assessment of LR and ANN based fault prediction techniques.....	59
<i>Bindu Goel, Yogesh Singh</i>	
Part 2: Advances in image processing	71
Chapter 1: A new scheme of image watermarking.....	73
<i>Chokri Chemak, Jean Christophe Lapayre, Mohamed Salim Bouhlef</i>	

Chapter 2:	Diffusing vector field of gray-scale images for image segmentation	87
	<i>X. D. Zhuang, N. E. Mastorakis</i>	
Chapter 3:	Image processing via synchronization in self-organizing oscillatory network	103
	<i>Eugene Grichuk, Margarita Kuzmina, Eduard Manykin</i>	
Chapter 4:	Feature based color stereo matching algorithm using restricted search.....	113
	<i>Hajar Sadeghi, Payman Moallem, S. Amirhassan Monadjemi</i>	
Chapter 5:	A novel edge preserving lossy image coder	121
	<i>Osslan Osiris Vergara Villegas, Manuel de Jes's Nandayapa Alfaro, Vianey Guadalupe Cruz Sánchez</i>	
Part 3: Modelling and simulation		129
Chapter 1:	Equivalent circuit extraction for passive devices	131
	<i>Yi-Ting Kuo, Hsueh-Yung (Robert) Chao, Yiming Li</i>	
Chapter 2:	Sleep quality differences according to a statistical continuous sleep model.....	141
	<i>A. G. Ravelo-García, F. D. Lorenzo-García, J. L. Navarro-Mesa</i>	
Chapter 3:	Simulation modeling in support of a European airspace study	151
	<i>John Morris, John James, James DeArmon, Kelly Connolly, Paula Mahoney</i>	
Chapter 4:	3D measurement system for car deformation analysis	163
	<i>András Rövid, Takeshi Hashimoto, Péter Várlaki</i>	
Chapter 5:	Adaptive MV ARMA identification under the presence of noise.....	177
	<i>Stylianios Sp. Pappas, Vassilios C. Moussas, Sokratis K. Katsikas</i>	
Chapter 6:	Planck's law simulation using particle systems	191
	<i>Douglas Farias Cordeiro, Marcos Aurélio Batista, Celia A. Zorzo Barcelos</i>	

Chapter 7: Simulation results, using shock absorbers with primary intelligence - VZN.....	199
<i>Adrian-Ioan Niculescu, Tudor Sireteanu, Daniel Dumitriu</i>	
Chapter 8: Pattern generation via computer.....	209
<i>Tatsuo Suzuki, Yasumichi Hasegawa</i>	
Chapter 9: 3D reconstruction: estimating depth of hole from 2D camera perspectives.....	225
<i>Muhammad Abuzar Fahiem, Shaiq A. Haq, Farhat Saleemi, Huma Tauseef</i>	
Chapter 10: Modeling diversity in recovery computer systems.....	235
<i>George Popov, Valeri Mladenov</i>	
Chapter 11: Cell automata models for a reaction-diffusion system	247
<i>Costin-Radu Boldea, Afrodita Balasa Boldea</i>	
Chapter 12: 3D reconstruction of solid models from 2D camera perspectives.....	253
<i>Muhammad Abuzar Fahiem</i>	
Chapter 13: DiProNN: VM-based distributed programmable network node architecture	265
<i>Tomáš Rebok</i>	
Chapter 14: A semantics of behavioral concepts for open virtual enterprises	275
<i>Mohamed Bouhdadi, Youssef Balouki</i>	
Chapter 15: A semantics of community related concepts in ODP enterprise language.....	287
<i>Mohamed Bouhdadi, Youssef Balouki</i>	
Chapter 16: Modeling of speech process including anatomical structure of vocal tract	299
<i>Zygmunt Ciota</i>	
Chapter 17: Carbon nanotube FET with asymmetrical contacts.....	307
<i>Alireza Kargar, Zoheir Kordrostami, Mohammad Hossein Sheikhi</i>	
Chapter 18: Optimizing prosthesis design by using virtual environments.....	313

Adrian Zafiu, Monica Dascălu, Eduard Franti, Annely-Mihaela Zafiu

Part 4: Multi-agent systems 321

- Chapter 1: A Maude based tool for simulating DIMA model..... 323
Noura Boudiaf, Lakhdar Sahbi
- Chapter2: Using two-level morphology to recognize derived
 agentive nouns..... 337
Arbana Kadriu
- Chapter 3: Towards an Agent Framework to support distributed
 problem solving..... 347
G.A. Mallah, Z.A. Shaikh, N.A. Shaikh
- Chapter 4: Building moderately open multi-agent systems: the
 HABA process 355
A. Garcés, R. Quirós, M. Chover, J. Huerta, E. Camahort

Part 5: Educational software and e-Learning systems..... 365

- Chapter 1: Using mathematics for data traffic modeling within an e-
 learning platform 367
Marian Cristian Mihăescu
- Chapter 2: Developing statistics learning at a distance using formal
 discussions 375
Jamie D. Mills
- Chapter 3: Distance teaching in the course of technology in Senior
 High School 385
Shi-Jer Lou, Tzai-Hung Huang, Chia-Hung Yen, Mei-Huang Huang, Yi-Hui Lui
- Chapter 4: Electronic exams for the 21st century 403
Antonios S. Andreatos
- Chapter 5: Effects of Orff music teaching method on creative
 thinking abilities..... 413
Rong-Jyue Fang, Hung-JenYang, C. Ray Diez, Hua-Lin Tsai, Chi-Jen Lee, Tien-Sheng Tsai

Chapter 6:	Knowledge application toward preparing engineering high school teachers	423
	<i>Kuo-Hung Tseng, Ming-Chang Wu, See-Chien Hou</i>	
Chapter 7:	Collaborative on-line network and culture exchange project	433
	<i>Rong-Jyue Fang, Hung-JenYang, C. Ray Diez, Hua-Lin Tsai, Chi-Jen Lee, Tien-Sheng Tsai</i>	
Chapter 8:	A study of the project on mobile device in education	443
	<i>Rong-Jyue Fang, Hung-Jen Yang, Hua-Lin Tsai, Chi-Jen Lee, Chung-Ping Lee, Pofen Wang</i>	
Chapter 9:	The solutions of mobile technology for primary school	455
	<i>Rong-Jyue Fang, Hung-JenYang, C. Ray Diez, Hua-Lin Tsai, Chi-Jen Lee, Tien-Sheng Tsai</i>	
Chapter 10:	A study of verifying knowledge reuse path using SEM	471
	<i>Hung-Jen Yang, Hsieh-Hua Yang, Jui-Chen Yu</i>	
Chapter 11:	A web-based system for distance learning of programming	483
	<i>V.N. Kasyanov, E.V. Kasyanova</i>	
Chapter 12:	New software for the study of the classic surfaces from differential geometry	495
	<i>Anca Iordan, George Savii, Manuela Pănoiu, Caius Pănoiu</i>	
Chapter 13:	Data requirements for detecting student learning style in an AEHS	505
	<i>Elvira Popescu, Philippe Trigano, Mircea Preda</i>	
Chapter 14:	Ontology-based feedback e-learning system for mobile computing	513
	<i>Ahmed Sameh</i>	
Part 6:	Information communication technologies	525
Chapter 1:	Potentiality of information technology system	527
	<i>Emmanouil Zoulias, Stavroula Kourtesi, Lambros Ekonomou, Angelos Nakulas, Georgios P. Fotis</i>	
Chapter 2:	A review of techniques to counter spam and spit	537

	<i>Angelos Nakulas, Lambros Ekonomou, Stavroula Kourtesi, Georgios P. Fotis, Emmanouil Zoulis</i>	
Chapter 3:	Sequential algorithms for max-min fair bandwidth allocation.....	549
	<i>Włodzimierz Ogryczak, Tomasz Śliwiński</i>	
Chapter 4:	A temporal variation in indoor environment.....	563
	<i>A. Jraifi, E.H Saidi, A. El Khafaji, J. El Abbadi</i>	
Chapter 5:	Workflow management for cross media publishing.....	573
	<i>Andreas Veglis, Andreas Pomportsis</i>	
Chapter 6:	Bayesian approach to improve the performance of P2P networks.....	583
	<i>Bertalan Forstner, Hassan Charaf</i>	
Chapter 7:	Deploying BitTorrent into mobile environments	597
	<i>Imre Kelényi, Bertalan Forstner</i>	
Chapter 8:	Numerical simulation for the Hopf bifurcation in TCP systems.....	609
	<i>Gabriela Mircea, Mihaela Muntean</i>	
Chapter 9:	Communications by vector manifolds	617
	<i>Guennadi Kouzaev</i>	
Chapter 10:	A computation study in mesh networks by scheduling problems	625
	<i>G. E. Rizos, D. C. Vasiliadis, E. Stergiou, E. Glavas</i>	
Chapter 11:	Serially concatenated RS codes with ST turbo codes over ring	633
	<i>P. Remlein, R. Orzechowski</i>	
Chapter 12:	Network performance monitoring and utilization measurement	643
	<i>Goran Martinovic, Drago Zagar, Drazen Tomic</i>	
Chapter 13:	Blind adaptive multiuser detection based on affine projection algorithm	651
	<i>Amin Khansefid, Ali Olfat</i>	
Chapter 14:	Multiple base station positioning for sensor applications...	663
	<i>M.Amac Guvensan, Z.Cihan Taysi, A.Gokhan Yavuz</i>	

Chapter 15: Considerations for the design of a general purpose wireless sensor node	675
<i>Z. Cihan Taysi, A. Gokhan Yavuz</i>	
Chapter 16: Performance comparison and improvement of wireless network protocols	689
<i>G. E. Rizos, D. C. Vasiliadis, E. Stergiou</i>	
Chapter 17: Modeling the erroneous behaviour of a sequential memory component with streams.....	699
<i>Walter Dosch</i>	
Chapter 18: FPGA implementation of PPM I-UWB baseband transceiver	711
<i>K. S. Mohammed</i>	
Chapter 19: WSN-based audio surveillance systems	723
<i>R. Alesii, G. Gargano, F. Graziosi, L. Pomante, C. Rinaldi</i>	
Chapter 20: Smart antenna design using multi-objective genetic algorithms	731
<i>I.O. Vardiambasis, N. Tzioumakis, T. Melesanaki</i>	
Part 7: Computer applications in modern medicine	737
Chapter 1: Intention to adopt the e-health services system in a bureau of health.....	739
<i>Hsieh-Hua Yang, Jui-Chen Yu, Hung-Jen Yang, Wen-Hui Han</i>	
Chapter 2: Visualization and clustering of DNA sequences	751
<i>Krista Rizman Žalik</i>	
Chapter 3: E-health service in Taiwan - The role of organizational innovativeness	763
<i>Hsieh-Hua Yang, Jui-Chen Yu, Hung-Jen Yang</i>	
Chapter 4: Analysis of heart sounds with wavelet entropy.....	775
<i>S. Bunluechokchai, P. Tosaranon</i>	
Chapter 5: Detection of mitral regurgitation and normal heart sounds	785
<i>S. Bunluechokchai, W. Ussawawongaraya</i>	

Chapter 6:	Wavelet entropy detection of heart sounds	793
	<i>T. Leeudomwong, P. Woraratsoontorn</i>	
Chapter 7:	Continuous wavelet transform analysis of heart sounds	803
	<i>P. Woraratsoontorn, T. Leeudomwong</i>	
Chapter 8:	Limitations of lung segmentation techniques.....	811
	<i>Nisar Ahmed Memon, Anwar Majid Mirza, S.A.M. Gilani</i>	
Chapter 9:	Segmentation of anatomical structures using volume definition tools	825
	<i>S. Zimeras, G. Karangelis</i>	
Chapter 10:	Shape analysis of heart rate Lorenz plots	837
	<i>France Sevšek</i>	
Chapter 11:	Human readable rule induction in medical data mining	845
	<i>MD Nor Ridzuan Daud, David Wolfe Corne</i>	
Part 8:	Computer applications in power systems.....	857
Chapter 1:	Wind energy production forecasting	859
	<i>G. Atsalakis</i>	
Chapter 2:	Operation management on autonomous power system	873
	<i>E.S. Karapidakis</i>	
Chapter 3:	A low-power portable H.264/AVC decoder using elastic pipeline	883
	<i>Yoshinori Sakata, Kentaro Kawakami, Hiroshi Kawaguchi, Masahiko Yoshimoto</i>	
Part 9:	Computational techniques and algorithms	891
Chapter 1:	Computational methods for detecting types of nonlinear dynamics	893
	<i>Mirela-Catrinel Voicu</i>	
Chapter 2:	The Equation Chapter 1 Section 1worm cutter tool profiling based on discrete surfaces representation	907
	<i>Virgil Teodor, Marian Cucu, Nicolae Oancea</i>	

Chapter 3:	Suggesting correct words algorithms developing in FarsiTeX	923
	<i>Youssef Rezvan, Majid Ghandchi, Farshad Rezvan</i>	
Chapter 4:	On the analytical and numerical computation in mechanical modeling.....	933
	<i>Marcel Migdalovici</i>	
Chapter 5:	Real genetic algorithm for traveling salesman problem optimization	941
	<i>A.R. Awad, I. Von Poser, M.T. Aboul-Ela</i>	
Chapter 6:	The COTECOMO: CONstructive Test Effort COst MOdel.....	953
	<i>Ljubomir Lazić, Nikos Mastorakis</i>	
Chapter 7:	A semantics of core computational model for ODP specifications.....	979
	<i>Youssef Balouki, Mohamed Bouhdadi</i>	
Chapter 8:	Computer simulation via direct modeling.....	991
	<i>Alex Soiguine</i>	
Chapter 9:	Image segmentation with improved artificial fish swarm algorithm.....	1003
	<i>Mingyan Jiang, Nikos E. Mastorakis, Dongfeng Yuan, Miguel Angel Lagunas</i>	
Chapter 10:	Propagation of elastic-plastic waves in bars	1011
	<i>Olga Martin</i>	
Chapter 11:	Shahyat algorithm as a clustering method	1023
	<i>Omid Khayat, Hamid Reza Shahdoosti</i>	
Chapter 12:	A modified electromagnetism-like algorithm based on a pattern search method	1035
	<i>Ana Maria A. C. Rocha, Edite M. G. P. Fernandes</i>	
Chapter 13:	Latency hiding and adaptability for parallel iterative algorithms	1043
	<i>Javier Roca, José Antonio Álvarez, Jose Jesús Fernández</i>	
Chapter 14:	Clustering the linearly inseparable clusters.....	1051
	<i>Hamid Reza Shahdoosti, Omid Khayat</i>	

Chapter 15: A top-down minimization algorithm for non-deterministic systems.....	1059
<i>Adrian Zafiu, Stefan Holban</i>	

Part 10: Natural phenomena and computer models 1067

Chapter 1: Multi-layer finite volume solution of wind induced basin flow.....	1069
<i>Saeed-Reza Sabbagh-Yazdi, Hamed Arabi, Nikos E Mastorakis</i>	
Chapter 2: NASIR heat generation and transfer solver for RCC dams	1079
<i>Saeed-Reza Sabbagh-Yazdi, Ali-Reza Bagheri, Nikos E Mastorakis</i>	
Chapter 3: Water hammer modeling using 2nd order Godunov finite volume method.....	1093
<i>Saeed-Reza Sabbagh-Yazdi, Ali Abbasi, Nikos E Mastorakis</i>	
Chapter 4: Simulation of wind pressure on two tandem tanks.....	1105
<i>Saeed-Reza Sabbagh-Yazdi, Farzad Meysami-Azad, Nikos E Mastorakis</i>	
Chapter 5: Tropical cyclone forecaster integrated with case-based reasoning.....	1115
<i>James N.K. Liu, Simon C.K. Shiu, Jane You</i>	

Part 11: Knowledge engineering, decision rules and data bases..... 1123

Chapter 1: Toward an understanding of knowledge reuse in on-line environment	1125
<i>Hsieh-Hua Yang, Jui-Chen Yu, Hung-Jen Yang</i>	
Chapter 2: Handling contradictions in knowledge discovery applications	1143
<i>David Anderson</i>	
Chapter 3: A cooperative distributed query-processing approach.....	1159
<i>Saeed K. Rahimi, Frank S. Haug</i>	
Chapter 4: Estimating join and projection selectivity factors	1173

	<i>Carlo dell'Aquila, Ezio Lefons, Filippo Tangorra</i>	
Chapter 5:	Decision rules: a metamodel to organize information	1185
	<i>Sabina-Cristiana Mihalache</i>	
Chapter 6:	A computerized solution for the financial diagnose of the SMEs	1197
	<i>Vasile Daniel Păvăloaia</i>	
Chapter 7:	DELP system: Tracking deadlocks and phantoms in databases	1209
	<i>Cristea Boboila, Simona Boboila, Constantin Lupsoiu</i>	
Chapter 8:	Concurrency control for multilevel secure distributed real-time databases	1217
	<i>Valuchandhar Vadivelu, S. Albert Rabara</i>	
Chapter 9:	A new framework for non-deterministic multivalued system minimization.....	1231
	<i>Adrian Zafiu, Ion Stefanescu</i>	
Part 12:	Web-based applications	1239
Chapter 1:	Thread pool-based improvement of the mean-value analysis algorithm.....	1241
	<i>Ágnes Bogárdi-Mészöly, Takeshi Hashimoto, Tihamér Levendovszky, Hassan Charaf</i>	
Chapter 2:	Security enhancements for web based applications.....	1255
	<i>Shivanand B. Hiremath, Sameer. S. Saigaonkar</i>	
Chapter 3:	Description and discovery of web applications in grid.....	1269
	<i>Serena Pastore</i>	
Chapter 4:	An approach to treat the user's preferences about personal data	1279
	<i>Sérgio D. Zorzo, Reginaldo A. Gotardo, Paulo R. M. Cereda, Bruno Y. L. Kimura, Ricardo A. Rios, Robson E. Grande</i>	
Chapter 5:	Ssort: Peak analysis for efficient string sorting.....	1287
	<i>Sunil Mallya, Kavi Mahesh</i>	

Chapter 6:	Parametric studies for the AEC domain using InteliGrid platform	1297
	<i>Matevž Dolenc, Robert Klinc</i>	
Chapter 7:	An auction-based resource allocation strategy with a proportional share model for economic-based grid systems.....	1305
	<i>Kuan-Chou Lai, Chao-Chin Wu, Jhen-Wei Huang</i>	

Part 13: Advances in computer science and applications 1311

Chapter 1:	Propagation in cylindrical inset dielectric guide structures	1313
	<i>Ioannis O. Vardiambasis</i>	
Chapter 2:	On the damage in nonlinear mesoscopic materials	1319
	<i>Veturia Chiroiu, Dan Dumitriu, Ana Maria Mitu, Daniel Baldovin, Cornel Secara</i>	
Chapter 3:	Diagnosis of robotized assembling systems.....	1325
	<i>C. Ciufudean, R. Serban, C. Filote, B. Satco</i>	
Chapter 4:	Open architecture systems for MERO walking robots control.....	1337
	<i>Luige Vladareanu, Radu Munteanu, Adrian Curaj, Ion N. Ion</i>	
Chapter 5:	Mass transfer analysis in the case of the EFG method	1345
	<i>Liliana Braescu, Thomas F. George, Stefan Balint</i>	
Chapter 6:	Information system architecture for data warehousing.....	1357
	<i>Francisco Araque, Alberto Salguero, Cecilia Delgado</i>	
Chapter 7:	Dimensional dynamics identification of reconfigurable machine tools	1365
	<i>Ciprian Cuzmin, Virgil Teodor, Nicolae Oancea, Vasile Marinescu, Alexandru Epureanu</i>	
Chapter 8:	Compaction-induced deformation on flexible substrate ...	1373
	<i>B. Punantapong</i>	
Chapter 9:	Robust and accurate visual odometry by stereo	1385
	<i>Aldo Cumani, Antonio Guiducci</i>	
Chapter 10:	Financial auditing in e-business environment.....	1395

L. D. Genete, Al. Ţugui

Chapter 11: Reputed authenticated routing ad-hoc networks protocol. 1407

Ahmed Sameh, Abdalla Mahmoud, Sherif El-Kassas

Chapter 12: Quantum mechanical analysis of nanowire FETs 1419

Alireza Kargar, Rahim Ghayour

Part 14: Software tools and programming languages 1425

Chapter 1: Impact of quality factors on the success of software development 1427

M. Sepehri, A. Abdollahzadeh, M. Sepehri, M. Goodarzi

Chapter 2: Sisal 3.2 language structure decomposition 1437

V.N. Kasyanov, A.P. Stasenko

Chapter 3: An embedded application for driverless metro train 1451

Vivek Kumar Sehgal, Nitin

Chapter 4: A post-mortem JavaSpaces debugger 1459

Jason Howarth, Edward Stow

Chapter 5: Stochastic numerical simulations using C# 1469

Edita Kolářová, Csaba Török

Chapter 6: Visualization and analyses of multi-dimensional data sets using OLAP- A case study of a student administration system 1475

Che-Chern Lin, Fun-Chi Hsu, Chun-Hung Wu, Chia-Jui Hsu, Chia-Hsun Wu, Chung-Ping Lee, Hung-Jen Yang

Chapter 7: Design of a computer-based triangle chess machine 1481

Che-Chern Lin, Chia-Jung Hsu, Ming-Jun Chan, Chih-Hung Chung, Chong-Cheng Hong, Hung-Jen Yang

Chapter 8: A taxonomy model for analyzing project-based learning using Analytical Hierarchy Process (AHP): A case study of design of a beer company website in an elementary school 1487

Chung-Ping Lee, Che-Chern Lin, Rong-Jyue Fang, Howard Lo, Ming-Chih Wu

Chapter 9: A framework of mixed model of a neural network and boolean logic for financial crisis prediction.....	1495
<i>Che-Chern Lin, Ghuan-Hsien Mao, Howard Lo, Hung-Jen Yang</i>	
Chapter 10: Chinese document clustering using self-organizing map-based on botanical document warehouse.....	1501
<i>Howard Lo, Che-Chern Lin, Rong-Jyue Fang, Chungping Lee, Yu-Chen Weng</i>	
Chapter 11: Linear discriminant analysis in Ottoman alphabet character recognition	1511
<i>Zeyneb Kurt, H. Irem Turkmen, M. Elif Karsligil</i>	
Chapter 12: User requirements engineering and management in software development.....	1519
<i>Zeljko Panian</i>	
Chapter 13: SSAF – A tool for active filter synthesis	1533
<i>Elena Doicaru, Lucian Bărbulescu, Claudius Dan</i>	
Part 15: Theoretical approaches in information science.....	1541
Chapter 1: On the complexity of the auxetic systems	1543
<i>Ligia Munteanu, Dan Dumitriu, Ștefania Donescu, Veturia Chiroiu</i>	
Chapter 2: New verification of reactive requirement for Lyee method	1551
<i>Osamu Arai, Hamido Fujita</i>	
Chapter 3: A tree derivation procedure for multivalent and paraconsistent inference.....	1567
<i>David Anderson</i>	
Chapter 4: On the feedback stabilization of matrix.....	1575
<i>JianDa.Han, Zhe.Jiang, YiWen.Zhao, YiYong.Nie</i>	
Chapter 5: Maxwell geometric dynamics	1597
<i>Constantin Udriște</i>	
Chapter 6: Fractional right ideals	1611
<i>Ulrich Albrecht</i>	

Chapter 7:	A distributed approach for concurrent password recovery of archived files.....	1619
	<i>Esra Celik, Z. Cihan Taysi, A. Gokhan Yavuz</i>	
Chapter 8:	Solutions for an integral-differential equation	1629
	<i>Nikos E. Mastorakis, Olga Martin</i>	
Chapter 9:	Development of genetic algorithms and C-method for optimizing a scattering by rough surface	1639
	<i>N. Lassouaoui, H. Hafdallah Ouslimani, A. Priou</i>	

Part 8

Computer applications in power systems

In order to meet the increasing demand for reliable design and expansion of power networks, the use of computers plays a leading role in the power industry's future R & D programs. Digital computers are already widely used in the analysis and design of large power networks. Hybrid computers offer a convenient alternative, especially where fast and accurate simulation of complex nonlinearities is required. Computer models can be used to simulate the changing states of electrical power systems. Such simulations enable the power engineer to study performance and predict disturbances. Computer models include mathematical background, algorithms and the basic tools needed to study complex power systems, their interaction and likely response to different types of network pathologies or disturbances. In this part we present some computer modeling techniques that constitute the framework of modern power system analysis. The papers include wind energy production forecasting, operation management on autonomous power system and a low-power portable H.264/AVC decoder using elastic pipeline.

Chapter 1

Wind energy production forecasting

G. Atsalakis

Department of Production Engineering and Management
Technical University of Crete, Chania, Greece
atsalak@otenet.gr

Abstract. This paper presents a forecasting system of short-term wind energy production. The system is a feedforward neural network that uses genetic algorithm as a learning method of forecasting the wind energy production of the following day. The data concerns four different wind power plants. The neural network forecasts are compared with the predictions made by a neural network that uses Quasi-Newton learning algorithm and are evaluated based on the four common statistical measures. Both models perform satisfactory as far as the prediction of the following day wind energy production is concerned.

Keywords. Wind power forecasting, wind energy, neural networks, genetic algorithms

1.1 Introduction

Wind energy is the energy source that contributes mostly to the renewable energy mix in Europe. The rapid expansion usage of wind energy within most countries requires the development of functional and accurate tools for wind energy production. The importance of wind power forecasting up to 48 hours ahead has been recognized by the wind plant operators, utilities, transmission system operators and other end-users. The short-range prediction (10-48 hours ahead) is required, if wind energy should become

a substitute for other energy sources, for example coal or gas. The accuracy requirements are not as high as for shorter ranges and depend on the electricity trading mechanisms. An important problem arises by the intermittency of the wind for the deployment of wind energy into the electricity networks.

To ensure grid security a Transmission System Operator needs today for each kilowatt of wind energy either an equal amount of spinning reserve or a forecasting system that can predict the amount of energy that will be produced from wind over a period of 1 to 48 hours. In the range from 5m/s to 15m/s a wind turbine's production increases with a power of three. For this reason, a Transmission System Operator requires accuracy for wind speed forecasts of 1m/s in this wind speed range. The major alternatives to wind energy forecasting are backup storage capacities and a strong inter-connectivity between the electrical grids. Both alternatives are associated with technical and economical restrictions, which can be barriers to the installation of larger amounts of wind energy. Forecasting can therefore be considered as a more efficient way of increasing wind energy penetration.

1.2 Related work

Many studies focus on predicting the wind speed rather than the power output. Traditionally, short-range forecasts have utilized on-site observations with various persistence based on statistical forecast models, including autoregressive time series techniques (Brown [7], Poggi [30], Weisser [36]). Other traditional methods are used by Youcef Ettoumi [35]. Giebel [13] surveys the literature on short-range wind speed and wind power forecasting. Similarly Sfetsos [34] compares various techniques applied in wind energy forecasting. Atsalakis [2, 3, 4, 5] presents neuro-fuzzy models (ANFIS) to energy forecasting. Also fuzzy theory has been applied by Chedid [8], Sanders [33] and Michalik [27].

Artificial neural networks are being applied in the energy forecasting issue by Datta [10], Khotanzad [20], Papalexopolous [29], Kretzschmar [21], Czernichow [9], Javeed Nizami [17], Rummeihart [31], Abdel-Aal [1], Mohandes [28], Kalogirou [18], Mandal [24] and Metaxiotis [26].

1.3 Theoretical approach

1.3.1 Neural networks

Artificial neural networks reproduce the behavior of the brain and nervous systems in a simplified computational form. They are constituted of highly interconnected simple elements, called artificial neurons, which receive information, elaborate them through mathematical functions and pass them to other artificial neurons. Particularly, in multilayer perceptron feed-forward networks the artificial neurons are organized in layers: an input layer, one or more hidden layers and an output layer [32, 14].

Neural networks may be considered as a "multivariate nonlinear nonparametric inference technique that is data driven and model free" [6]. In fact they are information-processing models designed to capture the underlying relationships among various data sets autonomously. They were firstly developed in the 1950s, and indeed the analogies to human intelligence are significant. Neural network systems are constructed along the same organizational principles as in the human brain, performing mathematical mapping from one domain or layer to another by using algorithmic techniques.

Neural networks will not only discover the relationships among the data sets and use these to predict the future values, but will use the actual results as they become known to adjust the formulas to predict more accurately in the years to come. Essentially, they learn from their mistakes. Over time, as they continue to accumulate data and feedback, the neural networks will retest and hone their formulas to achieve ever more accurate predictions.

There are several advantages in using Neural Networks. The most vital issue which needs to be mentioned is, the ability to learn from data and thus the potential to generalize, i.e. produce an acceptable output for previously unseen input data (important in prediction tasks). This even holds (to a certain extent) when input series contain low-quality or missing data. Another valuable quality is the non-linear nature of a neural network. Potentially a vast amount of problems may be solved. Furthermore no expert system is needed which makes the network extremely flexible to the changes in the environment. Only one has to retrain the system. Regarding downsides, the black-box-property first springs to mind. When relating one single outcome of the network to a specific internal decision turns out to be extremely difficult. Noisy data also reinforces the negative implications of establishing incorrect causalities, overtraining (or overfitting), which will damage generalization. Finally, a certain degree of knowledge in the sub-

ject is required as it is not trivial to assess the relevance of the chosen input series. In this study, one hidden layer is considered, since it is believed that this type of network can approximate any function [16, 23].

1.3.2 Quasi-Newton method for training the neural network

The non-linear optimization algorithms are based on minimizing the sum of squared errors in respect with the parameter set Ω (see Eq. 1.1):

$$\psi = \sum_{t=1}^{T-1} \{Q_t - \hat{\psi}_t\}^2 \quad (1.1)$$

where T is the number of observations in the simulation run, $\hat{\psi}_t$ is the approximated value of Q_t , conditional on information at time t .

To minimize a non-linear function, the parameter vector Ω must be randomly initialized, and then an iterating process is used on the coefficient set Ω until ψ is minimized. Such a process is based on the Quasi-Newton algorithm. Starting with the initial set of sum of squared errors $\psi(\Omega_0)$, based on the initial coefficient vector Ω_0 and a second order Taylor expansion is used to find $\psi(\Omega_1)$ (see Eq. 1.2):

$$\psi(\Omega_1) = \psi(\Omega_0) + \nabla_0(\Omega_1 - \Omega_0) + \frac{1}{2}(\Omega_1 - \Omega_0)' H_0(\Omega_1 - \Omega_0) \quad (1.2)$$

where ∇_0 is the gradient of the error function with respect to the parameter set Ω_0 and H_0 is the Hessian of the error function. To find the direction of the parameter set from iteration 0 to iteration 1, the error function $\psi(\Omega_1)$ in respect with $(\Omega_1 - \Omega_0)$ must be minimized (see Eq. 1.3):

$$(\Omega_1 - \Omega_0) = -H_0^{-1} \nabla_0 \quad (1.3)$$

The iteration is continued until the error function is minimized according to the tolerance value or when the maximum number of iterations is reached. The main disadvantage of this method is that one may not get results of global solution, but local or saddle point solution.

1.3.3 Genetic algorithm for training the neural network

The inclusion of Genetic Algorithm (GA) search techniques was undertaken for two reasons. The first relates to the potential GAs offer in terms of adaptiveness. The flexibility, robustness, and simplicity which GAs offers, renders them very attractive in that respect. The second reason stems from the difficulty in optimizing neural network applications. Genetic Algorithms by operating on entire populations of candid solutions on a parallel way are much less likely to get stuck at a local optimum. Since GAs search the fitness landscape stochastically, they are more likely than traditional techniques to estimate the true global optimum. In this paper, the optimization task is defined as an improvement of forecasting the ability of the network. Genetic algorithms are a class of probabilistic search techniques based on biological evolution [11, 12, 15]. The algorithms mimic biological evolution by applying a computationally simulated version of "survival of the fittest" [19].

A Genetic Algorithm starts with a population of random coefficient vectors. Then, two pairs of coefficients are selected, with a replacement. The best fitness of the coefficient vectors is given by the minimization of the sum of the squared errors. The winner of each pair has the best fitness. These two winning vectors are retained for 'breeding' purposes. The next step is the crossover, where the two parents 'breed' two children. Follow step is the mutation of the children. With some small probability, which decreases over time, each element or coefficient of the two children's vectors is subject to a mutation. Following the mutation operation, the four members of the 'family' engage in a fitness tournament. The children are evaluated by the same fitness criteria, which are used to evaluate the parents. The two vectors with the best fitness, whether parents or children, survive and pass to the next generation, whilst the two with the worst fitness value are extinguished. The above process is repeated, with parents returning to the population pool for possible selection again, until the next generation is populated [25].

1.4 Computational results

1.4.1 Parameter selection

The quantitative parameters used in this implementation were the minimum, the average, and the maximum speed of the wind, the minimum, the

average, the maximum speed of helix, the minimum the average and the maximum attributed force and the daily hours of operation.

Table 1.1. List of inputs

Input index	Symbol	Description of inputs
1	M_S_MIN	Minimum wind speed
2	M_S_AVG	Mean wind speed
3	M_S_MAX	Maximum wind speed
4	RPM_MIN	Minimum helix speed
5	RPM_AVG	Mean helix speed
6	RPM_MAX	Maximum helix speed
7	KW_MIN	Minimum attributed force
8	KW_AVG	Mean attributed force
9	KW_MAX	Maximum attributed force
10	H	Operation hours

They were then modified and acted upon scaled input values as received from the input layer. The scaled parameters can be considerate as a fluctuation rate. Table 1.1 presents the parameters that the model uses as inputs.

The processing layer (hidden) consists of 3 neurons which each one of them carries a different computation. The formulas for each neuron are composed from the modified parameters and are then interpreted through an activation function.

1.4.2 Network parameter issues

After comprehensive testing over both static and dynamic topologies, it was decided to go for a three-layered approach, as this was the most efficient computationally. The first layer was designated as the input layer where 10 input neurons would take 10 scaled input values of the series in the historic data.

The second layer is the hidden layer, which does all the computations, and it is comprised by three neurons. The three neurons in the hidden layer process the input in a parallel fashion to improve the predictions by using the sigmoid transfer function.

The third layer (the output layer) consists of one neuron with combinations of different weights on the inputs received from the hidden layer. The weighted mean of these inputs is the next predicted value and it is scaled back to give the exact value of the predictions.

The sigmoid transfer function is ideally used in this study rather than the gaussian. The sigmoid function acts as an output gate that can be opened

when it takes the value 1 or closed when it takes the value 0. Since the function is continuous, it is also possible for the gate to be partially opened (i.e. somewhere between 0 and 1). Models incorporating sigmoid transfer functions often help generalized learning characteristics and yield models with improved accuracy however the use of sigmoid transfer functions can also lead to longer training times. The gaussian transfer function significantly alters the learning dynamics of a neural network model. While the sigmoid function acts as a gate (opened, closed or somewhere in-between) for a neuron's output response, the gaussian function acts like a probabilistic output controller. Like the sigmoid function, the output response is normalized between 0 and 1, but the gaussian transfer function is more likely to produce the "in-between state". It would be far less likely, for example, for the neuron's output gate to be fully opened (i.e. an output of 1). Given a set of inputs to a neuron, the output will normally be some type of partial response. That means that the output gate will open partially. Gaussian based networks tend to learn quicker than sigmoid counterparts, but can be prone to memorization.

1.4.3 Data series partitioning

The data concerns four wind power plants that are established in a Greek island. The data sample of each plant contains 365 daily time series observations. One typical method for training a network is to firstly separate the data series into two disjoint sets: the training set and the test set. On this aspect 80% of the data is used for training and the remaining 20% for the evaluation of the models.

The network is trained directly on the training set and its ability to forecast is measured on the test set. A network's generalization ability indirectly measures how well the network can deal with unforeseen inputs, in other words, inputs on which it was not trained in. A network that produces high forecasting errors on unforeseen inputs, but low errors on training inputs, is said to have an over fit of the training data. Overfitting occurs when the network is blindly trained to a minimum in the total squared errors and is based on the training set. A network that has over fit the training data is said to have poor generalization ability.

The data is transformed before is fed to the neural network for training and testing. It is necessary to have all training targets normalised between 0 and 1, as the output neuron is restricted to a signal of values of only 0 or 1. As a result all inputs have also been normalised to the 0 to 1 range. It is accepted that by conducting this normalising process the characteristics of the training process are enhanced [28]. The data is scaled in order to facili-

tate the nonlinear estimation process and to avoid the underflow or overflow problems. Helge Petersohn's scaling function is used to scale the data from zero to one, denoted $[0, 1]$, [25]. To compare the models estimations with the actual values the data is transformed again in no scaling status.

1.4.4 Learning and training

In this model, wind energy production depends on a set of current variables that are described above in the network's parameters. Two types of neural models were created. One type has been trained with genetic algorithms. The genetic algorithms do not involve taking gradients or second derivatives and is a global and evolutionary search process. The population size is set to 50 and the number of generation is set to 20.

The other type was developed in order to compare the first type, and uses the Quasi-Newton optimization method for training. Using this training method there is a strong danger of getting stuck in a local rather than a global minimum. Furthermore the Hessian matrix may fail to invert, or become 'near-singular' leading to imprecise or even absurd results, for the coefficient vector of the neural network. The number of iterations is set to 20,000.

Learning is accomplished by using the two above mentioned supervised methods. In supervised learning, the network is provided with the actual outputs which in turn are compared with the produced output values, so that the network can note the difference and begin to adjust itself or learn from certain actions. The weights of the arcs between the nodes are adjusted based on the above learning algorithms and aim to decrease the difference between produced and actual values. The network then receives the new input and repeats the procedure, seeks further reduction. After training, weights associated with the best performance on the training set were selected and were applied to the evaluation of the models.

1.4.5 Out of sample evaluation

This section focuses on the out-of-sample forecasting ability of the two competing models. Each of the forecasting models described in the last section are estimated by in-sample data. Once the training process of the models is completed, they are ready to predict the series, which they are trained upon. The prediction process is very similar to the training process, except from the evaluation and the parameter adjustment steps. The prediction horizon here is one step ahead forecasts. Since the prediction being one step ahead it is implied that the models will predict the next day's

energy production. The out-of-sample forecasting is referred to a sample of 73 daily observations, which is the testing sample of the models. On the basis of the estimated coefficients of the models, one-step-ahead forecasts were generated from both models.

At this stage, the relative performance of the models is measured by four frequently used measures. To name them: root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE) and mean squared error (MSE) [22]. Table 1.2 reports the out of sample forecasting errors of the four plants that are calculated by the neural network that has been trained with the genetic algorithms.

Table 1.2. NN forecasts using genetic algorithm as learning method

	Plant 1	Plant 2	Plant 3	Plant 4	Average
RMSE	2.090	2.587	2.745	2.963	2.596
MAE	1.686	1.925	2.259	2.211	2.020
MAPE	15.772	18.285	24.143	22.239	20.105
MSE	5.671	6.692	13.99	8.787	5.287

Table 1.3 reports the out of sample forecasting errors of the same four plants that are calculated by the neural network which in turn have been trained with the Quasi-Newton method. A plant by plant comparison of the two different types of neural networks when the above learning approaches are used, they yields slightly different results.

Table 1.3. NN forecasts using Quasi –Newton as learning method

	Plant 1	Plant 2	Plant 3	Plant 4	Average
RMSE	2.631	2.397	2.911	3.504	2.860
MAE	2.085	1.727	2.253	2.750	2.203
MAPE	17.716	16.182	20.418	32.011	21.581
MSE	6.926	5.747	8.477	12.284	8.358

When the neural network model is using the genetic algorithms approach it has lower RMSE, in the three of the four plants, than those of the neural network model when applying the Quasi-Newton approach. Also it has lower MAE in two of the four plants, lower MAPE in three of the four plants and lower MSE in two out of four plants. Slight but clear improvement on the forecasts, in terms of the four plants' mean error, produces the neural network that uses the genetic algorithm approach. It gives mean values lower than in all of the above statistical performance measures (RMSE, MAE, MAPE and MSE). This implies that the forecasting values

given by a neural network that uses the genetic algorithms approach are in general more accurate than the forecasting that is based on the Quasi-Newton approach.

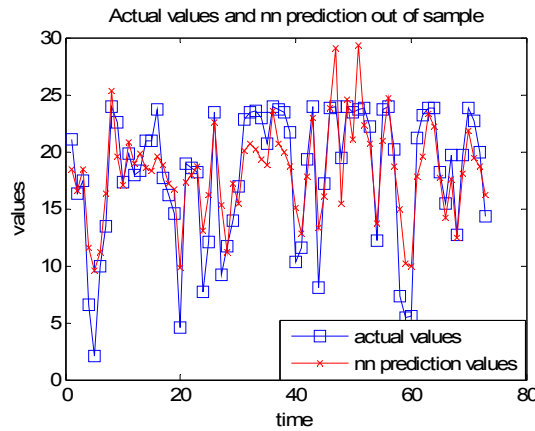


Fig. 1.1. Genetic algorithm approach

Figure 1.1 illustrates the actual values and the out of sample wind power production forecasting of the neural network that is trained with genetic algorithm for the first plant. In many cases the two lines coexist.

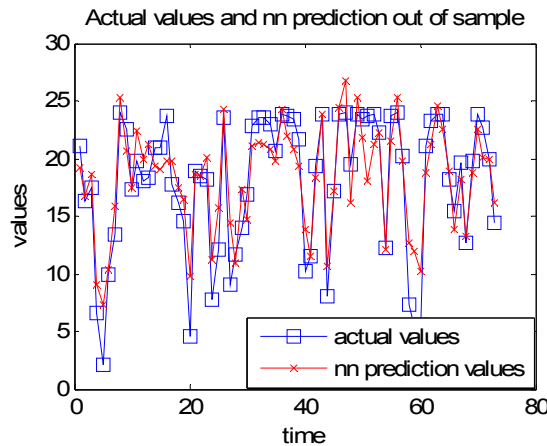


Fig. 1.2. Quasi-Newton approach

Figure 1.2 illustrates the out of sample forecasts of the neural network, which is trained by the Quasi-Newton algorithm, as well as the actual values of the first plant. As one can identify from the graph, the models pro-

duced a high level of accuracy in the prediction of 24 hours average in wind power production.

1.5 Conclusion and further research

To sum up this paper studied the predictive performance of neural networks that applied genetic algorithms for training, using data from four real-word wind power plants. The results were compared with a neural network that uses the Quasi-Newton method for training. The data contains 365 daily observations. It was scaled and the 80% of the samples were used for training, and the remaining 20% were used for testing the models. Four statistical measures were calculated to estimate the predictive performance of the models. The average error of the four plants, for all of the statistical measures shows the best performance of the neural network model that uses genetic algorithms as a training method. Both models produce satisfactory forecasts of the wind energy production of the following day.

Neural networks do not give a clear explanation of how they reach to a conclusion. In case that there is necessary to be known how a particular conclusion is made, further research can be following using neuro-fuzzy prediction methods.

References

1. Abdel-Aal RE, Al-Garni AZ, Al-Nassar YN (1997) Modeling and forecasting monthly electric energy consumption in Eastern Saudi Arabia using adductive networks. *Energy* 22:911–921
2. Atsalakis G, Ucenic C (2006) Electric load forecasting by Neuro-fuzzy approach. In: WSEAS International Conference on Energy and Environmental Systems, Evia, Greece
3. Atsalakis G, Ucenic C (2006) Forecasting the electricity demand using a Neuro-fuzzy approach versus traditions methods. *Journal of WSEAS Transactions on Business and Economics* 3: 9-17
4. Atsalakis G, Ucenic C (2006) Forecasting the wind energy production using a Neuro-fuzzy model. *Journal of WSEAS Transactions on Environment and Development* 2:6 823-829
5. Atsalakis G, Ucenic C, Plokamakis G (2005) Forecasting of electricity demand using Neuro-fuzzy (ANFIS) approach. In: International Conference on NHIBE, Corfu, Greece
6. Azoff E (1994) Neural network time series forecasting of financial markets. Wiley, New York

7. Brown BG, Katz RW, Murphy AH (1984) Time series models to simulate and forecast wind speed and wind power. *Journal of Climate and Applied Meteorology* 23:1184-1195
8. Chedid R, Mezher T, Jarrouche C (1999) A fuzzy programming approach to energy resource allocation. *International Journal of Energy Res* 23: 303-317
9. Czernichow T, Germond A, Dorizzi B, Caire P (1995) Improving recurrent network load forecasting. In: *Proceedings IEEE International Conference ICNN'95 Perth (Western Australia)*, 2:899-904
10. Datta D, Tassou SA (1997) Energy management in supermarkets through electrical load prediction. In: *Proceedings 1st International Conference on Energy and Environment, Limassol (Cyprus)*, 2:493-587
11. Davis L (1991) *Handbook of genetic algorithms*. Van Nostrand Reinhold, New York
12. Goldberg D E (1989) *Genetic algorithms in search optimization and machine learning*. Reading MA: Addison-Wesley
13. Giebel G (2003) The state-of-the-art in short-term prediction of wind power, Deliverable Report D1.1, Project Anemos, Available online at anemos.cma.fr/modules.php?name=Downloads&dop=viewdownload&cid=3
14. Hagan M T, Demuth HB, Beale M (1996) *Neural network design*. PWS Publishing Company, Boston
15. Holland J H (1975) *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press
16. Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. *Neural Networks* 2:359-366
17. Javeed Nizami S, Al-Garni AG (1995) Forecasting electric energy consumption using neural networks. *Energy Policy* 23:1097-1104
18. Kalogirou SA (2000) Applications of artificial neural networks for energy systems. *Applied Energy* 67:17-35
19. Kingdon J (1997) *Intelligent systems and financial forecasting*. Springer-Verlag, Berlin
20. Khotanzad A, Hwang RC, Maratukulam D (1993) Hourly Load Forecasting by Neural Networks. *Proceedings of the IEEE PES Winter Meeting, Ohio*
21. Kretzschmar R, Eckert P, Cattani D, Eggimann F (2004) Neural network classifiers for local wind prediction. *Journal of Applied Meteorology* 43: 727-738
22. Makridakis S, Wheelwright SE, McGee VE (1983) *Forecasting: Methods and Applications*. Wiley, New York
23. Maier H R, Dandy G C (2000) Neural Networks for prediction and forecasting of water re- resources variables: a review of modeling issue and applications. *Environmental Modeling & sources Software* 15:101-124
24. Mandal JK, Sinha AK, Parthasarathy G (1995) Application of recurrent neural network for short term load forecasting in electric power system. In: *Proceedings IEEE International Conference ICNN'95, Perth (Western Australia)*, 5:2694-2698
25. McNelis D P (2005) *Neural Networks in Finance: Gaining predictive edge in the market*. Elsevier Academic Press

26. Metaxiotis K, Kagiannas A, Ashounis D, Psarras J (2003) Artificial intelligence in short term electric load forecasting: a state-of-the-art survey for the researcher. *Energy Convers Management* 44:1525–1534
27. Michalik G, Khan ME, Bonwick WJ, Mielczarski W (1997) Structural modeling of energy demand in the residential sector: the use of linguistic variables to include uncertainty of customer's behavior. *Energy* 22:949–958
28. Mohandes AM, Rehman S, Halawani TO (1998) A neural network approach for wind speed prediction. *Renewable Energy* 13:345–354
29. Papalexopoulos AD, Shangyou H, Peng TM (1994) An Implementation of Neural Network Based Load Forecasting Models for the EMS. *IEEE Transactions on Power Systems* 9(4)
30. Poggi P, Muselli M, Notton G, Cristofari C, Louche A (2003) Forecasting and simulating wind speed in Corsica by using an autoregressive model. *Energy Convers Management* 44:3177–3196
31. Rumelhart DE, Hinton BE, Williams RJ (1986) Learning Internal Representations by Error Propagation. In: Rumelhart DE, McClelland JL (eds) *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, Cambridge, Mass: MIT Press
32. Rosenblatt F (1958) The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65:386–408
33. Sanders I, Batty WJ, Hagino K (1993) Supply of and demand for a resource: fuzzy logistical optimization technique. *Applied Energy* 46:285–302
34. Sfetsos A (2000) A comparison of various forecasting techniques applied to mean hourly wind speed time series. *Renew Energy* 21:23–35
35. Youcef Ettoumi F, Sauvageot H, Adane AEH (2003) Statistical bivariate modeling of wind using first-order Markov chain and Weibull distribution. *Renew Energy* 28:1787–1802
36. Weisser D (2003) A wind energy analysis of Grenada: an estimation using the 'Weibull' density function. *Renew Energy* 28:1803–1812

Chapter 2

Operation management on autonomous power system

E.S. Karapidakis

Technological Educational Institute of Crete, 3 Romanou str, 73133, Chania, Greece

Abstract. The running developments in the field of power systems have as a result, the maximization of complexity and their marginal operation with regard to their dynamic security. This becomes more perceptible in autonomous power systems. Consequently, in the modern energy environment, the use of enhanced operation management and monitoring programs is judged necessary. The results that were carried out in the present paper were developed via concretization of algorithms, which were incorporated in an implemented operation and planning management system. The presented algorithms have as object to improve the level of combination between unit commitment, economic dispatch and dynamic security assessment.

Keywords. Power Systems, Dynamic Security Assessment and Energy Management System

2.1 Introduction

Energy Management System (EMS) is one of the main Control Center's programs. The high processing power, which is provided by modern computers, creates the requirement for significant fast and reliable energy management programs in power system's control centers, [Jamnicksy L.]. An EMS also includes dynamic security assessment algorithms (DSA) for real time performance [Dy-Liacco T.E.], which considered as critical sup-

port factor for rational planning and power system operation. Specifically, any EMS is separated in two basic categories, depending on their implementation time:

- The real time planning and operation programs, where the load and corresponding wind power production forecasting (when such possibility exists), supplies the necessary data entry for economic dispatch (ED) and unit commitment (UC) algorithm integration. Then, DSA algorithms provides the safety level of given profile under pre-selected contingencies, in order a final and optimal operation decision to be retrieved.
- The short-term planning and system operation programs, where off-line the system administrator analyze and estimate the next most optimal possible operation system profiles via different scripts.

The results that were carried out in the present paper were developed via concretization of algorithms, which were incorporated in an implemented operation and planning management system. The presented algorithms have as object to improve the level of combination between unit commitment, economic dispatch and dynamic security assessment.

2.2 Algorithm implementation

The main modules of the implemented program are related either to real time operation management, or to short-term power system planning, as it is shown in Figure 2.1. Specifically, the following sub-algorithms were integrated and examined.

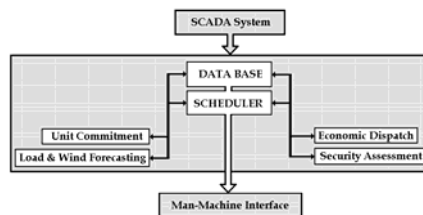


Fig. 2.1. EMS Functions & Procedures

2.2.1 Load and wind power forecasting

The mathematic methods for load demand forecasting, and in certain cases, for wind power generation forecasting, are distinguished in three basic categories:

- Time series methods, where load demand curves follow given seasonal, monthly, weekly and daily periodicity, [Box G.E., Jenkins G.M.].
- Regression based methods, where load demand forecasting calculated by linear combination of previous hours, [Gross G., Galiana F.D.].
- Computational intelligence based methods, through supervised learning, where large number of historical data are used for artificial intelligence architectures training, [Park D.C. et al., Lee K.Y. et al., Highly D.D. and Hilmes T.J., Kariniotakis G. et al.].

Table 2.1. Parameter description

Symbol	Description
N	Total number of power units
T	Time
D^t	Load demand
R^t	Required spinning reserve
g^t	Power generation
$g^{\min}$	Minimum power generation
$g^{\max}$	Maximum power generation
u^t	Decision of unit i operation
$C_i(g_i)$	Fuel cost function of unit i
$r(g_i)$	Spinning reserve function of unit i
s_i	Start-up cost of unit i
$r_{\max}$	Maximum spinning reserve

2.2.2 Unit commitment

Electrical power generation Units Commitment (UC) problem is a complex optimization problem with lot of restrictions, [Haili Ma et al., Androutsos A. et al., Kazarlis K. et al.]. Specifically, the mathematic expression is given as:

$$\min \sum_{t=0}^{T-1} \sum_{i=1}^N u_i^t C_i(g_i^t) + (1 - u_i^{t-1}) u_i^t s_i \quad (2.1)$$

with the following restrictions, as load demand on the examined time horizon (2.2), the required spinning reserve (2.3) and the technical minimum and maximum limits of units generation (2.4).

$$\sum_{i=1}^N g_i^t = D^t \quad (2.2)$$

$$\sum_{i=1}^N r(g_i^t) \geq R^t \quad (2.3)$$

$$g_i^{\min} \leq g_i^t \leq g_i^{\max} \quad (2.4)$$

The previous equations represent the problem of optimal economically power units generation profile in power systems. In following Table 2.1, all the parameters are individual described.

2.2.3 Economic dispatch

Economic analysis of each conventional generation units is based on the following general economic parameters:

- Hourly Fuel Consumption (Kcal/h): the hourly fuel consumption is given as function of generating power P of the corresponding unit.
- Hourly Fuel Cost (Euro/h): the hourly fuel cost results from the hourly fuel consumption and the corresponding cost, according to the relation:

$$F = \frac{kcal}{h} \cdot \frac{Euro}{kcal} = \frac{Euro}{h} \quad (2.5)$$

- Differential Hourly Fuel Consumption $d(Kcal/h)/d(MW)$: the differential hourly fuel consumption represents the difference of hourly consumption fuel ($d(Kcal/h)$), which results from produced power change ($d(MW)$), as it appears in:

$$\frac{d(kcal/h)}{d(MW)} \quad (2.6)$$

- Differential Production Cost (Euro/MWh): the differential production cost results from the differential hourly consumption and the corresponding cost, according to the relation:

$$\frac{\frac{d(kcal/h)}{d(MW)} \cdot \frac{Euro \cdot 10^6}{10^6 \cdot kcal}}{10^6 \cdot kcal} = \frac{Euro}{MWh} \quad (2.7)$$

An energy system, which allocates a significant number of power generation units, with different technical and economical characteristics, the total production cost varies sensibly, according to production combinations of the available generators. Accordingly, economic dispatch of power generation units is a classic optimization problem with restrictions, [Gooi H.B.

et al., Contaxis G. et al.]. The mathematic expression of the previous optimization problem is:

$$\min(\sum_{i=1}^N C_i(P_i)) \quad (2.8)$$

with the following restrictions, as load demand on the examined time horizon (2.9), the required spinning reserve (2.10) and the technical minimum and maximum limits of units generation (2.11).

$$\sum_{i=1}^N P_i = D + L \quad (2.9)$$

$$\sum_{i=1}^N P_i \geq P \quad (2.10)$$

$$P_i^{\min} \leq P_i \leq P_i^{\max} \quad (2.11)$$

The previous equations (2.8), (2.9), (2.10) and (2.11) represent the optimal economic dispatch problem of power generation for the running load demand. In Table 2.2, all the corresponding parameters are individually described.

Table 2.2. Parameter description

Symbol	Description
N	Total number of power units
D	Load demand
R	Required spinning reserve
P_i	Power generation of unit i
$P^{\min}$	Minimum power generation
$P^{\max}$	Maximum power generation
$C(P_i)$	Fuel cost function of unit i
R_i	Spinning reserve of unit i

Fuel consumption curve of each unit is given in the majority of cases from a second degree polynomial function:

$$C(P_i) = a \cdot P^2 + b \cdot P + c \quad (2.12)$$

Previous relation is used in the proposed algorithm, while it is considered as significant analytic and representative to the real production cost of the corresponding units.

2.2.4 Dynamic security assessment

A reliable dynamic secure operation monitoring requires the analysis of tens or even hundreds of possible operating profiles (steady & transient), in desirable and practically feasible computational time, [Kumar A.B.R. et al.].

The mathematic methods for dynamic security assessment of a power system under pre-selected disturbances (as short-circuits and unit's trip) are distinguished in two categories:

- Conventional Mathematic Methods, where the results are given via analytic recurrent mathematic functions or corresponding mathematic models, which describe in detail the examined system. However, the analytic mathematic solution of this complex problem requires prohibitory a lot of time, taking into account the modern EMS real time operation, [Ejebe G.C. et al.].
- Expert systems and computational intelligence based methods, where the previous experience of operators, combined with considerably large number of historical data, is used for AI architectures training, [Pecas Lopes J.A., Hatziaargyriou N.D.], in order satisfactorily fast and precise DSA results to be achieved.

2.3 Operation management

A short-term generation planning and power system operation program was developed, receiving into account the technical restrictions and the economic characteristics of each generation unit. Initially, the program determines an available unit commitment for a given load demand curve, while the selected generators set points are recalculated via individual economic dispatch algorithms. Then, the dynamic safety level of the proposed operation profile is estimated under various pre-selected transient disturbances, in order an acceptance or a redefinition of new security margins to be retrieved. More precisely, the main program processes are:

1. 24-hours Economic Unit Commitment.
2. Dynamic Security Assessment of the previous Unit Commitment.
3. New Unit Commitment with new Dynamic Security Criteria.
4. Final Dynamic Security Level Verification.

The software package (algorithms & interface) has been implemented with Visual Basic (version 6) in multi document interface, while UC and DSA modules have been developed independently, as dynamic link libra-

ries (dlls), in order to give easy replacement. The database is a relational database (RDBMS) via dynamic access object (DAO 3.6).

2.3.1 Data entry

In following Figure 2.2, a data entry sample of load demand curve and wind power generation are depicted.

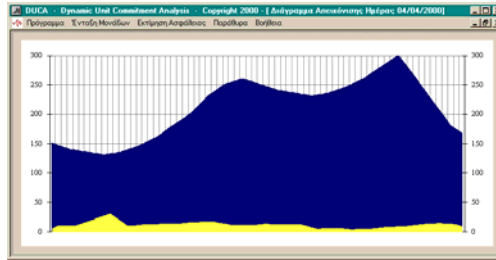


Fig. 2.2. Load and Wind Power Data Entry

2.3.2 Unit Commitment

Under selected load demand curve and the corresponding wind power production, the UC algorithm provides the most optimal calculated system generation profile.

The technical characteristics (min/max), cost curves, start-up costs and fuel consumptions, as well as the response type of each conventional generation unit, contribute to corresponding unit set points determination.

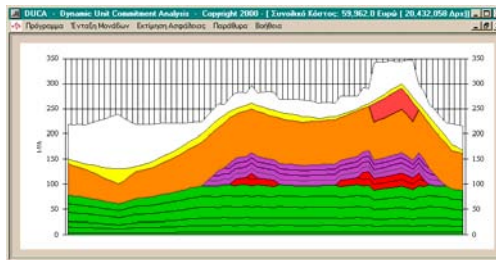


Fig. 2.3. Unit Commitment Monitoring

One more algorithm parameter that influences considerably the generation dispatch profile is the desirable spinning reserve. In Figure 2.3, UC results for the examined time horizon (24 hours), is presented, receiving into account all the previous data.

According to previous Figure 2.3, load demand is mainly covered by the “base” power units, as six steam units (green) and the combined cycle unit (orange). Continuously, high loads are covered by the four “fast” diesel units (mauve), while the main daily peaks are served by the available gas units (red).

Part of total load, as it was already reported, covers the selected wind power production (yellow) for the examined time period, while the spinning reserve (white) is calculated according to the proposed operation profile. The total cost of the proposed dispatch, with 10% spinning reserve limit, is calculated in 59,962 €.

2.3.3 Dynamic Security Assessment

The previous UC results provide the data entry of dynamic security level estimation algorithm under pre-selected disturbances of examined 24-hour schedule. In Figure 2.4, the level of dynamic safety, in parallel with load demand and wind power production, is presented.

Algorithms results are combined in one total indicator line (light blue), which is presented as percentage (%) of absolute security level, giving with this the way a picture of power system safety level in case of several contingences. More concretely, the current indicator results as medium term from the classification of examined functional points (0 unsafe and 1 safe) via five decision trees of pre-selected common disturbances.

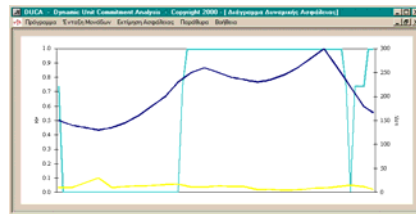


Fig. 2.4. Dynamic Security Level Monitoring

2.3.4 Preventive Security Assessment

Unit commitment module is executed once again, receiving as additional entry the dynamic security restrictions, which were exported by DSA algorithm, [Karapidakis E.S. and Hatziaargyriou N.D.]. In following Figure 2.5, final preventive security assessment results are plotted, where all the corresponding diagrams of the examined power system for short-term planning and operation are in one screen.

The improvement of dynamic security level is confirmed by safety index. As it is shown in Figure 2.5, it becomes easily obvious the difference between the previous security level without restrictions and the final proposed profile.



Fig. 2.5. Final Results Screen

2.4 Conclusion

From the start of electrical energy systems, human operators (system dispatcher) had the whole responsibility of power generation and distribution control. During eighties, Nathan Cohn (IEEE Computer Application in Power, January 1988) described a precocious energy management application of power generation and distribution control. Since then, the programs that are used in power systems control centers present significant improvement, based on the modern computer's increasing processing power, [Dy-Liacco T.E., Fred I. Denny].

Present paper presents a real time energy management program for short-term operation planning, which was initially applied in Crete's autonomous power system. The program calculations efficiency is judged satisfactory enough, receiving into account the evaluations results in real operation conditions.

References

1. Androutsos A, Papadopoulos M (1999) Logistics modeling and its real time implementation on large isolated power system with high wind penetration. EWEC 99, Nice, France

2. Box GE, Jenkins GM (1976) Time series analysis forecasting and control. San Francisco: Holden-Day
3. Contaxis G, Vlachos A (1999) Constrained optimal power flow in electrical energy grids with large integration of dispatchable wind energy and independent wind power producers. EWEC 99, Nice, France
4. Dy-Liacco TE (1988) System security: the computer's role. IEEE Spectrum, pp 45-50
5. Dy-Liacco TE (2002) Control Centers are here to stay. IEEE Computer Applications in Power 15:18-23
6. Ejebe GC, Jing C, Waight JG, Vittal V, Pieper G, Jamshidian F, Hirsch P, Sobajic D (1998) Online Dynamic Security Assessment in an EMS. IEEE Computer Applications in Power 11:43-47
7. Fred I Denny (2002) Prospective on Computer Applications in Power. IEEE Computer Applications in Power 15:24-29
8. Gooi HB, Mendes DP, W.Bell KR, Kirschen DS (1999) Optimal scheduling of spinning reserve. IEEE transactions on Power Systems 14:1485-1490
9. Gross G, Galiana FD (1987) Short term load forecasting. In Proc. IEEE, 75:1558-1573
10. Haili Ma, Shahidehpour SM (1999) Unit commitment with transmission security and voltage constraints. IEEE transactions on Power Systems, 14:757-764
11. Hatziaargyriou ND (1998) Dynamic security assessment of isolated power systems with increased wind power integration. Group 38, Pref. Subject 3, 37th Session, CIGRE, Paris
12. Highly DD, Hilmes TJ (1993) Load forecasting by ANN. IEEE Computer Applications in Power 6:10-15
13. Jamnicky L (1996) EMS network security applications of the future. IEEE Computer Applications in Power, vol 13
14. Karapidakis ES, Hatziaargyriou ND (2002) On-line preventive dynamic security of isolated power systems using decision trees. IEEE Transactions on Power Systems 17:297-304
15. Kariniotakis G, Matos M, Miranda V (1999) Assessment of benefits from advanced load and wind forecasting in Autonomous Power Systems. EWEC 99, Nice, France
16. Kazarlis K, Bakirtzis A, Petridis V (1996) A genetic algorithm solution to the unit commitment problem. IEEE Transaction on Power Systems 11:83-90
17. Kumar ABR, Brandwajn V, Ipakchi A, Rambabu Adapa (1998) Integrated Framework for Dynamic Security Analysis. IEEE Transactions on Power Systems 13(3):816-821
18. Lee KY, Cha YT, Park JH (1992) Short term load forecasting using an artificial neural network. IEEE Trans. PAS 7:124-131
19. Park DC, El Sharkawi MA, Marks RJ, Atlasm LE, Damborg J (1991) Electric load forecasting using an artificial neural network. IEEE Trans. PAS 6:442-448
20. Pecas Lopes JA (1998) Application of neural network based stability assessment tools to an operational environment of large autonomous power systems. Group 38, Pref. Subject 1, 37th Session, CIGRE, Paris

Chapter 3

A low-power portable H.264/AVC decoder using elastic pipeline

Yoshinori Sakata, Kentaro Kawakami, Hiroshi Kawaguchi, Masahiko Yoshimoto

Graduate School, Kobe University, Kobe, Hyogo, 657-8507 Japan
Email: {yoshi, kawakami, kawapy, yosimoto}@cs28.cs.kobe-u.ac.jp

Abstract. We propose an elastic pipeline architecture that can apply dynamic voltage scaling (DVS) to a dedicated hardware, and implement the elastic pipeline to a portable H.264/AVC decoder LSI with embedded frame buffer SRAM. A supply voltage and operating frequency are decreased by a feedback-type voltage/frequency control algorithm. In a portable H.264/AVC decoder, embedded SARM can be utilized as frame buffer since the frame buffer is not so large that an external DRAM is required. In the proposed pipeline architecture, the power in the embedded SRAM and even in a local bus connecting with the frame buffer SRAM can be controlled by DVS. We carried out simulation in the 320×180 pixels baseline profile and 320×240 pixels mail profile. The total power reduction in 320×180 pixels and 320×240 pixels are 30% and 31%, respectively.

Keywords. Dynamic voltage scaling, Elastic pipeline, Embedded sram, H.264/AVC, Memory bandwidth

3.1 Introduction

Dynamic voltage scaling (DVS) is a low-power technique that controls an operating frequency and a supply voltage on an LSI, according to an appli-

cation workload. DVS is well utilized on general-purpose processors to achieve both high peak performance and low average power [1, 2].

Figure 3.1 shows a relationship between an operating frequency and a power, with DVS and without DVS (= clock gating). In DVS, if a workload does not need a high operating frequency, we can choose a combination of a lower operating frequency and a lower supply voltage. Due to this power optimization, the power becomes lower than the conventional scheme without DVS, when a workload is low. If the maximum performance is instantaneously needed, the highest supply voltage and highest operating frequency are utilized so that DVS can accommodate the peak performance.

In a case of an application to hardwired logic circuits for real-time processing, there are a few problems; a dedicated hardware is often built with pipeline architecture for high performance. Considering the likely worst-case workload, the starting time of a pipeline process is segmented into the worst-case execution cycles (WCEC). Thus, the required operating frequency is uniquely fixed, and there is no room to apply DVS in the hardwired logic circuits.

To realize DVS in hardwired logic circuits, we propose an elastic pipeline architecture. Depending on characteristics of input data, this architecture can conserve process cycles in the pipeline operation. The slack time is exploited for DVS, which achieves lower power in hardwired logic circuits.

The rest of this paper is organized as follow. Section 3.2 mentions the conventional pipeline architecture. Section 3.3 describes the proposed elastic pipeline architecture for DVS in hardwired logic circuits. In Section 3.4, we exhibit the experimental results of the proposed architecture. Section 3.5 summarizes this paper.

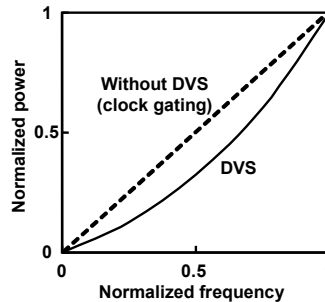


Fig. 3.1. Relationships between power and frequency in DVS and clock gating

3.2 Conventional pipeline architecture

Figure 3.2 illustrates a timing diagram of the conventional pipeline architecture. The WCEC is the maximum number of execution cycles required for one pipeline process. A gray area in the figure shows the number of processing cycles that a pipeline stage processes a datum. A hatching area means common idle cycles in a pipeline process after all pipeline stages were completed. Considering the worst-case workload, a starting time of each pipeline process is fixed to the WCEC in the conventional pipeline architecture. Hence, all the pipeline stages have to idle until the next starting time even if all the pipeline stages finish earlier than the WCEC.

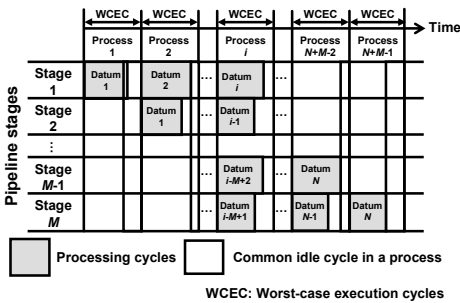


Fig. 3.2. Timing diagram of the conventional pipeline

3.3 Elastic pipeline architecture

3.3.1 Concept

We propose the elastic pipeline architecture as the solution to the issue of the conventional pipeline architecture [3]. Figure 3.3 (a) and (b) shows a concept and a timing diagram of the proposed elastic pipeline architecture. After each stage in the elastic pipeline was completed, it sends a completion signal to the pipeline controller. As soon as the pipeline controller collects all the completion signal from all the pipeline stages, each pipeline stage proceed to the next pipeline process with the start signal.

In the proposed architecture, the common idle cycles are built up in pipeline processes, and become a lump of time. As illustrated in Figure 3.3 (b), a pipeline process in the elastic pipeline requires less time than the conventional pipeline since the common idle cycles are put off. Thereby,

the elastic pipeline architecture produces the slack time, ΔH , compared to the conventional pipeline architecture.

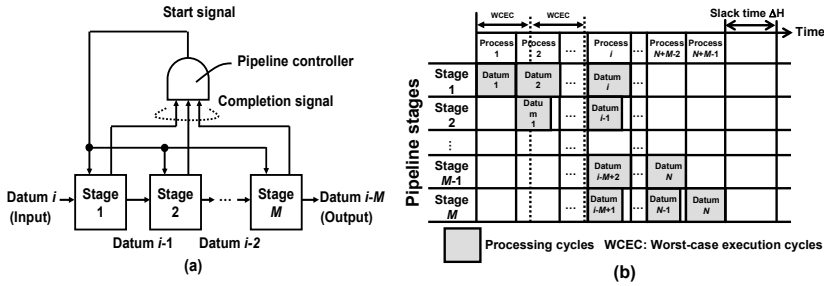


Fig. 3.3. Proposed elastic pipeline architecture: (a) concept and (b) timing diagram

3.3.2 Feedback-type voltage/frequency control algorithm

For DVS in the proposed elastic pipeline, a supply voltage and an operating frequency are changed by a feedback-type voltage/frequency control algorithm as illustrated in Figure 3.4 [4]. In an H.264 codec, data are processed in every single macro block (MB: 16×16 pixels). In this algorithm, a frame is divided into some slots; a set of MBs are assigned to a slot. The first and second slots are always processed with the maximum frequency ($= f$ in Figure 3.4). However, these slots are potentially completed earlier since the elastic pipeline reduces the number of processing cycles. Now, pay attention to the third slots, where the slack time, ΔH , is left. Even considering a voltage/frequency transition time, T_{td} , the third slot has twice as long time as T_{slot} (a processing time for a slot), which allows the third slot to be processed at an half of f . Note that a real-time operation is guaranteed in this feedback algorithm. As described, we prepare the two kinds of operating frequencies, f and $f/2$ in this study [3, 4].

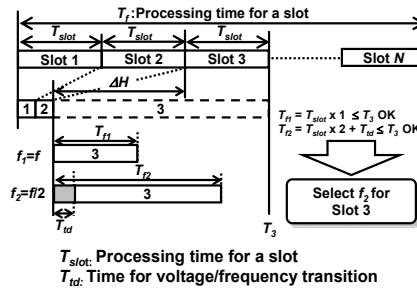


Fig. 3.4. Feedback-type voltage/frequency control algorithm

3.3.3 Architecture

To estimate the effect of the power reduction in the proposed pipeline architecture, we designed a H.264 decoder architecture as shown in Figure 3.5. SRAM utilized for reference images is embedded on a chip. The external DRAM is used as bit stream buffer, if the resolution is very large as HDTV. In this case, a supply voltage and operating frequency should be fixed since it is preferable that the DRAM interface operates at a fixed supply voltage and operating frequency for compatibility with other hardware cores. But in this study, we handle small resolution which is used for the portable product. So SRAM which utilized for bit stream buffer is embedded. We can adapt DVS for not only decoder core but SRAM and local bus connecting with the frame buffer SRAM.

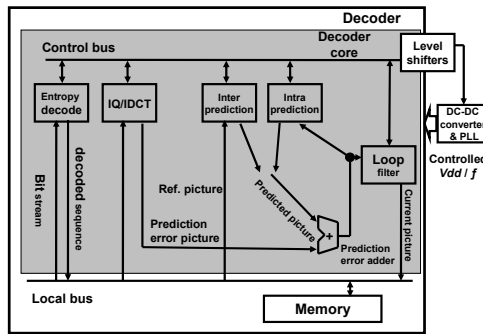


Fig. 3.5. H.264 decoder block diagram

3.4 Experimental results

3.4.1 Test sequence

Assuming portable H.264 image sequences, we handle two kinds of resolution (320×180 pixels and 320×240 pixels). As test sequences, we chose six sequences: Bus (BUS1), Cheer (CHER), Flower (FLOW), Foreman (FORE2), Football (FTBL), Girl (GIRL). Then, we encoded the six test sequences under the configurations in Table 3.1 (a) to prepare the test sequences: The baseline profile with 320×180 pixels complies with the Japanese portable television standard, and the main profile with 320×240 is adopted by Sony PSP [5].

Table 3.1. (a) Encoding configuration, (b) Simulation parameter

(a)		
Frame size (Resolution)	320 × 180 pixels	320 × 240 pixels
# of test sequence	6 sequences	6 sequences
Profile	Baseline profile	Main profile
Frame rate	15	30
# of reference frames	1	2
Reference software	JM9.6[6]	JM9.6[6]
(b)		
Frame size (Resolution)	320 × 180	320 × 240
# of slots	75	30
Operating frequency (MHz)	6.75/3.38	15.84/7.92
Supply voltage (V)	0.62/0.6	0.8/0.63
SRAM (Mbits)	2	4
WCEC	1760	1760
# of logic gates	600329	600329

Table 3.1 (b) illustrates the simulation parameters. The respective supply voltage and operating frequency are prepared for the two kinds of resolutions. The capacities of the embedded SRAMs are 2 M bits and 4 M bits, respectively. The SRAM capacity of the 320 × 240 pixel main profile is twice as large since the number of reference frames is two. However, note that the WCECs and the numbers of logic gates are equal between the two kinds of resolutions. In other words, the operating frequencies are different, but the sizes of the decoder cores are the same.

3.4.2 The optimum number of slots per frame

Since the elastic pipeline architecture reduces the processing cycle, we can apply DVS to the decoder LSI. In this subsection, the optimum number of slots is discussed.

Figure 3.6 illustrates the simulation result of the relationship between the power of the decoder core and the number of slots, using the BUS1 sequence. The power reduction factor depends on the number of slots. In this simulation, the transition time is assumed to be 50 μs [1]. The baseline profile with 320 × 180 pixels has the power minimum when the number of slices is 75. On the other hand, the optimum number of slots is 30 in the main profile with 320 × 240 pixels.

If the number of slots is smaller than the optimum point, the power reduction drastically becomes worse. For instance, if there are merely several slots, there are few chances to make the operating frequency and the

supply voltage lower, which causes high power consumption. Alternatively, if there are many slots, there are many chances to change the operating frequency and supply voltage. The voltage/frequency transition time, however, becomes longer. The power reduction gradually becomes worse with the increase of the slice number.

3.4.3 Power saving

As well as the decoder core, we can apply DVS to the embedded frame buffer SRAM and the local bus connecting to the SRAM. Figure 3.7 (a) shows the respective power reduction factors in the local bus, the embedded SRAM, and the decoder core. In the case of the frame size of 320×180 pixels, the proposed elastic pipeline reduces the powers by 7%, 25%, and 41%, respectively, in the local bus, the embedded SRAM, and the decoder core. In the case of the frame size of 320×240 pixels, the respective factors are 38%, 27%, and 33% in the resolution of 320×180 pixels.

The overall power reduction is 30% and 31% on average, in the two kinds of resolutions, as shown in Figure 3.7 (b).

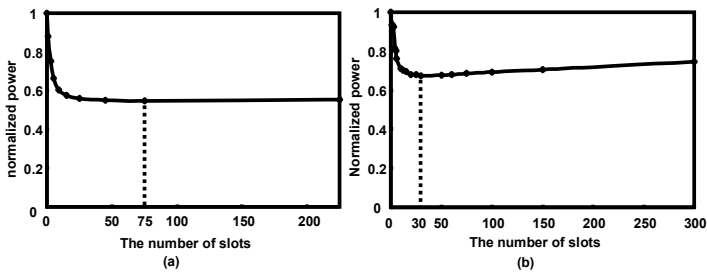


Fig. 3.6. The number of slots in a frame vs. power using the test sequence “Bus”: (a) 320 x 180 pixels and (b) 320 x 240 pixels

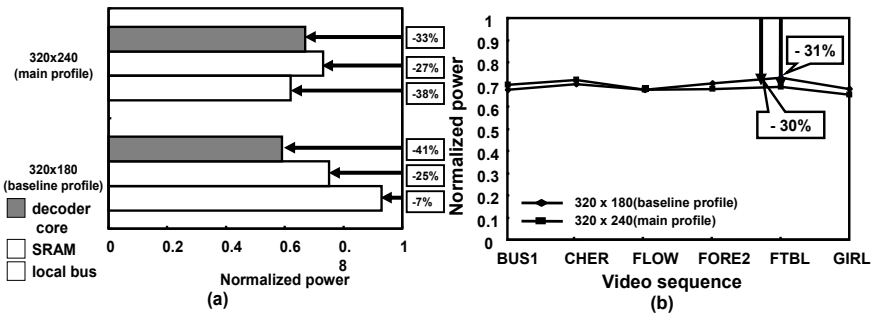


Fig. 3.7. Power reduction ratio: (a) decode core, embedded SRAM, local bus, (b) overall decoder

3.5 Summary

We proposed the elastic pipeline architecture that can apply DVS to a hardwired circuit. We implemented a H.264 decoder LSI, and controlled the embedded frame buffer SRAM and the local bus connecting to the embedded SRAM with DVS, as well as the decoder core.

We verified that the proposed elastic pipeline reduces the power on the H.264 decoder LSI by 7% in the local bus, by 25% in the frame buffer SRAM, and by 41% in a decoder core in a 320×180 pixel baseline profile. In a case that 320×240 pixel main profile, the power is reduced by 38%, 27%, and 33% in the local bus, the frame buffer SRAM, and the decoder core, respectively. The total power reductions in the baseline profile and the mail profile pixels are 30% and 31%, respectively.

References

1. Nowka KJ, Carpenter GD, MacDonald EW, Ngo HC, Brock BC, Ishii KI, Nguyen TY, Burns JL (2002) A 32-bit PowerPC system-on-a-chip with support for dynamic voltage scaling and dynamic frequency scaling. *IEEE J. Solid-State Circuits* 37(11):1441-1447
2. Kawakami K, Kanamori M, Morita Y, Takemura J, Miyama M, Yoshimoto M (2005) Power-minimum frequency/voltage cooperative management method for VLSI processor in leakage-dominant technology era. *IEICE Trans. Fundamentals* E88-A(12):3290-3297
3. Kawakami K, Kuroda M, Kawaguchi H, Yoshimoto M (2007) Power and memory bandwidth reduction of an H.264/AVC HDTV decoder LSI with elastic pipeline architecture. In: *Proceeding of Asia and South Pacific Design Automation Conference (ASP-DAC)*
4. Kawaguchi H, Shin Y, and Sakurai T (2005) μ ITRON-LP: power-conscious real-time OS based on cooperative voltage scaling for multimedia applications. *IEEE Trans. Multimedia* 7(1):67-74
5. <http://manuals.playstation.net/document/jp/psp/current/video/filetypes.html>
6. Joint Video Team (JVT) of ISO/IEC MPEG&ITU-T VCEG (2003) ISO/IEC 14496-10.

Part 9

Computational techniques and algorithms

In this part we consider some computational techniques, tools and algorithms for solving different problems. We presents theoretical and philosophical discussions, pseudocodes for algorithms, and computing paradigms that illustrate how computational techniques can be used to solve complex problems, simulate and explain natural phenomena, and possibly allow the development of new computing technologies. The topics include computational methods for detecting types of nonlinear dynamics, the worm cutter tool profiling based on discreet surfaces representation, suggesting correct words algorithms developing in FarsiTeX, analytical and numerical computation in mechanical modeling, real genetic algorithm for traveling salesman problem optimization, the constrictive test effort cost model, a semantics of core computational model for ODP specifications, computer simulation via direct modeling, image segmentation with improved artificial fish swarm algorithm, propagation of elastic-plastic waves in bars, Shahyat algorithm as a clustering method, a modified electromagnetism-like algorithm based on a pattern search method, latency hiding and adaptability for parallel iterative algorithms, clustering the linearly inseparable clusters and a top-down minimization algorithm for non-deterministic systems. The results can serve an interdisciplinary community which needs to use mathematical techniques in the development of computational analysis and engineering methodology, in order to better solve real life problems such as described above. It also aims to bridge the subject areas of mathematics, classical computer science, engineering, and future generation computational analysis.

Chapter 1

Computational methods for detecting types of nonlinear dynamics

Mirela-Catrinel Voicu

Faculty of Economic Sciences, West University of Timișoara, ROMANIA,
mirela.voicu@fse.uvt.ro

Abstract. In this paper we present implementation methods used to detect the dynamics displayed by nonlinear determinist discrete systems. Our study focuses on attractors. The most complex type of behavior for a nonlinear system is that of chaotic dynamics. It is known that the Lyapunov exponents are a tool used to establish the type of behavior in nonlinear dynamics. For this reason, in this paper we also present an implementation method used in order to obtain the Lyapunov exponents. In order to exemplify our study, we will present some numerical results for a certain system of the second order. Our implementation methods can be used for any nonlinear determinist discrete system and for any order of the system.

Keywords. Implementation methods, nonlinear systems, chaos, attractors

1.1 Introduction

In the study of nonlinear dynamics, situations occur where analytical tools alone are not sufficient. However, in many cases we can use numerical simulation to detect the type of dynamics. All numerical algorithms are based on the mathematical theory. Here are some important aspects: the computer performance, the software used for implementation and the implementation methods. All these must imply a very short time to detect the

results. If the results are obtained quickly, many simulations can be made, thus leading to a good conclusion to the study.

The aim of this paper is to emphasize the importance of implementation methods of computational study. As an example, we present some results obtained for a nonlinear determinist discrete dynamical system. In this paper we focus on chaotic behavior.

1.2 Mathematical framework presentation

1.2.1 Model presentation

We consider the following model:

$$s_t = \left(\frac{(c + \alpha)b}{1 + \beta(e^{s_{t-1}} - 1)^2} + (1 - \alpha)b \right) s_{t-1} + \left(\frac{-cb}{1 + \beta(e^{s_{t-1}} - 1)^2} \right) s_{t-2} \quad (1.1)$$

where $\alpha > 0$, $\beta > 0$, $b \in (0, 1)$, $c > 1$, $t \geq 0$, $s_t \in R$ and e^{s_t} is the exchange rate at the moment t .

The model presented in equation (1.1) is a model proposed in [2] and describes the exchange rate evolution.

The vectorial form of equation (1.1) is given by the following expression:

$$(s_t, s_{t+1}) = F(s_{t-1}, s_t) \quad (1.2)$$

where $F : R^2 \rightarrow R^2$, with $F(x, y) = (F_1(x, y), F_2(x, y))$, $F_1(x, y) = y$

$$\text{and } F_2(x, y) = \left(\frac{(c + \alpha)b}{1 + \beta(e^y - 1)^2} + (1 - \alpha)b \right) y + \left(\frac{-cb}{1 + \beta(e^y - 1)^2} \right) x.$$

In our numerical study we will use both forms. For certain values of parameters (α , β , b and c) and the initial condition, we have obtained some qualitative results (see [6]), but in many situations the analytical tools alone are not sufficient. For this reason, in order to detect different system behavior types, we have made numerical simulations. Fixing the values of parameters (α , β , b and c) and the initial condition, using software applications, we can generate an unlimited number of iterations of the system and

a graphical representation for their values. When the number of iterations increases, generally, we can observe the trajectory tendency.

On this study stage, we want to make some observations:

- The value ranges of parameters and of the initial condition cover sets with an infinite number of values, and a numerical study can be made only for a limited number of such values. The selection (for simulation) of certain parameters and initial condition values is influenced by different reasons (e.g., the value signification for the phenomenon which is modelled by the system).
- Using numerical simulations, we can easily detect only attractors (e.g., in the case of a periodical repulsive cycle, we must have the initial conditions on the cycle, and this is a practically impossible).

We can classify the different attractors: attracting fixed point, attracting periodical- p cycle, quasiperiodic attractor and strange attractor.

The sensitive dependence on initial condition is one of the most essential aspects for identifying the chaos. We recall that the sensitive dependence on initial conditions means that two trajectories starting very close together will rapidly diverge from each other. The strange attractor is associated with a chaotic state of time evolution and is characterized by the sensitive dependence on initial conditions.

A measure of the average rate of exponential divergence exhibited by a chaotic system is given by the Lyapunov exponents of the system; the positivity of such exponents can suggest the presence of chaos.

The Lyapunov exponents λ_1, λ_2 are given by

$$\{e^{\lambda_1}, e^{\lambda_2}\} = \lim_{n \rightarrow \infty} \left\{ \text{eigenvalues of} \left(\prod_{t=0}^{n-1} J(F(s_t, s_{t+1})) \right)^{\frac{1}{n}} \right\} \quad (1.3)$$

where $J(F(s_t, s_{t+1}))$ represents the Jacobian matrix of the function F . For a period- p point the Lyapunov exponents are given by

$$\{e^{\lambda_1}, e^{\lambda_2}\} = \left\{ \text{eigenvalues of} \left(\prod_{t=0}^{p-1} J(F(s_t, s_{t+1})) \right)^{\frac{1}{p}} \right\} \quad (1.4)$$

We recall now that for an attracting period- p cycle the Lyapunov exponents are negative; in case of a bifurcation point, at least one Lyapunov exponent is zero; for a limit cycle one Lyapunov exponent is zero and the

others are negative and for a chaotic behavior the highest Lyapunov exponent is positive while the sum of the all Lyapunov exponents is negative.

When system (1.2) displays a chaotic behavior, in order to compute the Lyapunov exponents, we use the method proposed in [1], based on the Householder QR factorization.

1.2.2 The Householder QR Factorization for a matrix $A \in R^2 \times R^2$

We recall the Householder QR Factorization (see [7]). For $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$

we choose the vector $x = \begin{pmatrix} a_{11} \\ a_{21} \end{pmatrix}$. The Householder vector according to x is

$$v = x - \|x\|_2 \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \text{where} \quad \|x\| = \sqrt{a_{11}^2 + a_{21}^2}. \quad \text{We obtain} \quad v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} a_{11} - \sqrt{a_{11}^2 + a_{21}^2} \\ a_{21} \end{pmatrix}.$$

$$H = I - \frac{2}{v^T v} v v^T, \quad \text{where} \quad v^T v = \begin{pmatrix} a_{11} - \sqrt{a_{11}^2 + a_{21}^2} & a_{21} \end{pmatrix} \begin{pmatrix} a_{11} - \sqrt{a_{11}^2 + a_{21}^2} \\ a_{21} \end{pmatrix} = \begin{pmatrix} a_{11} - \sqrt{a_{11}^2 + a_{21}^2} \end{pmatrix}^2 + a_{21}^2 \quad \text{and} \quad v v^T = \begin{pmatrix} a_{11} - \sqrt{a_{11}^2 + a_{21}^2} \\ a_{21} \end{pmatrix} \begin{pmatrix} a_{11} - \sqrt{a_{11}^2 + a_{21}^2} & a_{21} \end{pmatrix}.$$

$$\begin{pmatrix} a_{11} - \sqrt{a_{11}^2 + a_{21}^2} & a_{21} \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} a_{11} - \sqrt{a_{11}^2 + a_{21}^2} \end{pmatrix}^2 & a_{21} \begin{pmatrix} a_{11} - \sqrt{a_{11}^2 + a_{21}^2} \end{pmatrix} \\ a_{21} \begin{pmatrix} a_{11} - \sqrt{a_{11}^2 + a_{21}^2} \end{pmatrix} & a_{21}^2 \end{pmatrix}.$$

We obtain $H = \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix}$, where

$$H_{11} = 1 - \frac{2}{\begin{pmatrix} a_{11} - \sqrt{a_{11}^2 + a_{21}^2} \end{pmatrix}^2 + a_{21}^2} \begin{pmatrix} a_{11} - \sqrt{a_{11}^2 + a_{21}^2} \end{pmatrix}^2$$

$$H_{12} = -\frac{2}{\begin{pmatrix} a_{11} - \sqrt{a_{11}^2 + a_{21}^2} \end{pmatrix}^2 + a_{21}^2} a_{21} \begin{pmatrix} a_{11} - \sqrt{a_{11}^2 + a_{21}^2} \end{pmatrix}$$

$$H_{21} = -\frac{2}{\left(a_{11} - \sqrt{a_{11}^2 + a_{21}^2}\right)^2 + a_{21}^2} a_{21} \left(a_{11} - \sqrt{a_{11}^2 + a_{21}^2}\right)$$

$$H_{22} = 1 - \frac{2}{\left(a_{11} - \sqrt{a_{11}^2 + a_{21}^2}\right)^2 + a_{21}^2} a_{21}^2.$$

The matrix H has the property $H = H^T = H^{-1}$ because $HH = \left(I - \frac{2}{v^T v} vv^T\right)^2 = I - \frac{4}{v^T v} vv^T + \frac{4}{v^T v} vv^T = I$ implies $H = H^{-1}$.

The property $H^T = H^{-1}$ means that the matrix H is an orthogonal matrix.

The matrix HA is upper triangular because

$$\begin{aligned} HA_{21} = & -\frac{2}{\left(a_{11} - \sqrt{a_{11}^2 + a_{21}^2}\right)^2 + a_{21}^2} a_{21} \left(a_{11} - \sqrt{a_{11}^2 + a_{21}^2}\right) a_{11} + \\ & \left(1 - \frac{2}{\left(a_{11} - \sqrt{a_{11}^2 + a_{21}^2}\right)^2 + a_{21}^2} a_{21}^2\right) a_{21} - 2a_{21}a_{11}^2 + 2a_{21}a_{11}\sqrt{a_{11}^2 + a_{21}^2} + \\ & \left(a_{11} - \sqrt{a_{11}^2 + a_{21}^2}\right)^2 a_{21} + a_{21}^3 - 2a_{21}^3 = 0. \end{aligned}$$

If we choose $Q = H$ and $R = HA$ we obtain $A = HHA = QR$, this means that the matrix A is decomposed into an orthogonal matrix Q and an upper triangular matrix R .

If we denote $J_i = J(F(s_{i-1}, s_i))$, $A_1 = J_1, A_i = J_i Q_{i-1}$ for $i = \overline{2, n}$, then $\prod_{i=1}^n J_i = Q_n R_n \dots R_1$. It is shown, see [1], that the diagonal elements λ_{jj}^n , $j=1, 2$ of the upper triangular matrix product $R_n \dots R_1$ obtained from this algorithm verify the relation: $\lim_{n \rightarrow \infty} \ln |\lambda_{jj}^n| = \lambda_j$.

If r_{jj}^i are the diagonal elements of the matrix R_i , then $\lambda_j = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n \ln |r_{jj}^i|$.

1.3 Implementation methods for numerical simulations

1.3.1 Implementation method used to detect the behavior type of system (1.1)

In our computational study we are concentrated on some objectives:

1. to determine the behavior type for certain cases;
2. to obtain images for attractors;
3. to calculate the Lyapunov exponents;
4. to obtain animation.

The time necessary to obtain the results is very important for our objectives. In order to detect the behavior type of the equation (1.1), we will generate a number of iterations.

In the case in which the trajectory moves in time towards an attractor (fixed point, periodical cycle, limit cycle or chaotic attractor), the iterations number, which is necessary to observe the behavior type, is different from a case to another. However, in the case of a fixed point or a periodical cycle (with a small period) this number is small, but for other situations, this number can be considerably higher (some hundred thousands or even greater).

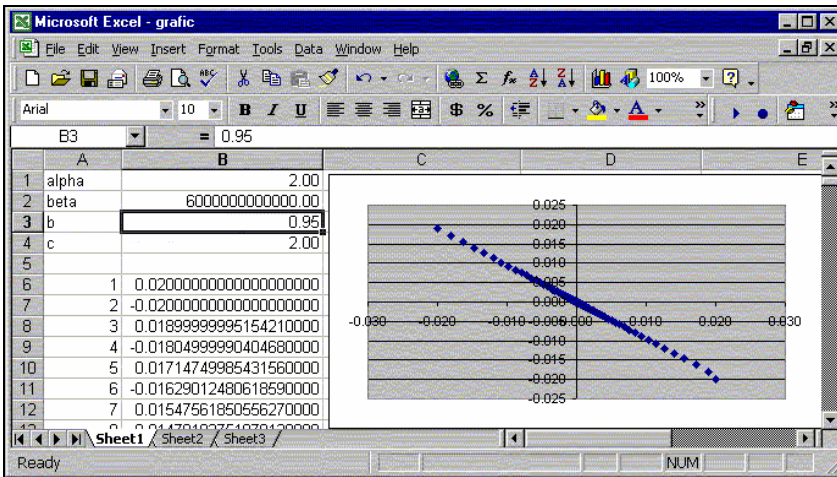


Fig. 1.1. The first 4000 points from the trajectory

In such a study, it is important to have simultaneously an image for the generated iterations, because in the moment in which we observe a tendency we can stop generating other iterations.

Moreover, in the case in which the trajectory moves in time towards an attractor, generally, the first iterations are not necessary in the graphical representation of the attractor. For example, in Figure 1.1 we have the image for the first 4000 points of a trajectory (in R^2). In Figure 1.2, we have the image for the points corresponding to the iterations (200 000 – 204 000).

From the observations presented before, we propose the following way of implementation:

- We generate a fixed number of iterations (for example, 20 000) and their graphical representation.
- If the image suggests a certain behavior type, we stop generating other iterations.
- If this is not the case, we will generate the following group of iterations (here we generate groups of 20 000 iterations – in fact, we will generate some groups with the same number of iterations, which makes the implementation easier). In this moment, we will also destroy the last group.

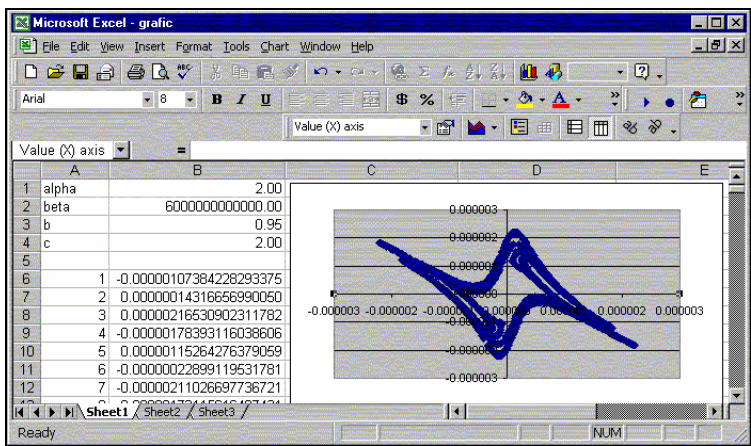


Fig. 1.2. Some points (from a trajectory) which provide the image of an attractor

For the last generated group of iterations, we will repeat the same algorithms.

This implementation manner enables the successive observation of an unlimited number of points of the trajectory (see Figure 1.3).

In order to provide such an implementation, we propose to work (for example) in *Excel* using macros.

In the cells *B1: B4* we introduce the values of the parameters and in the cells *B6: B7* we introduce the initial condition (see Figure 1.1).

In the cell *B8* we use the following *Excel* formula:

$$= ((B4+B\$1) * B\$3 / (1+B\$2 * (EXP(B7) - 1) ^ 2) + (1 - B\$1) * B\$3) * B7 - B4 * B\$3 / (1+B\$2 * (EXP(B7) - 1) ^ 2) * B6$$

and we copy this formula in the domain *B9:B20005*.

In order to obtain an image for the graphical representation of these points of the trajectory, we make a chart in the space (s_t, s_{t+1}) (see Figure 1.2). Here, we use 20 000 cells, in which the initial condition is changed with the two last iterations from the group, and we will repeat this change until we can observe a tendency in dynamics. In order to obtain the results quickly, we also use macros (which automatically execute these changes of values).

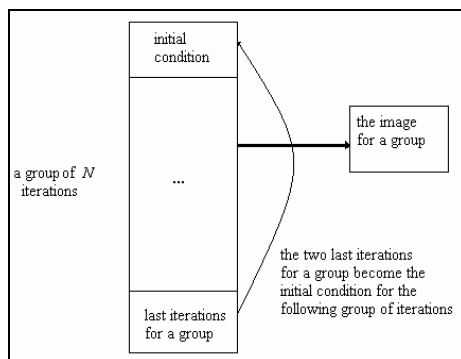


Fig. 1.3. Generating group of iterations

```

Sub initial_condition_changing()
  For i=1 To 10
    Application.Goto Reference:="R20005C1"
    Range("B20004:B20005").Select
    Selection.Copy
    Application.Goto Reference:="R1C1"
    Range("B6").Select
    Selection.PasteSpecial Paste:=xlValues, Opera-
    tion:=xlNone, SkipBlanks:= _
    False, Transpose:=False
    Range("E4").Select
  Next i
End Sub

```

In the case in which we observe a periodical- p cycle, theoretically we cannot be sure of the result, because we have a numerical approximation. However, we can calculate the Lyapunov exponents, implementing the definition (see equation (1.4)) and if the result is confirmed, we can arrive to a conclusion. In the case in which the attractor seems to be chaotic, we suggest to verify first if there is a sensitive dependence on the initial conditions (see Figures 1.4 and 1.5), and if so, to calculate the Lyapunov exponents using the method proposed in [1].

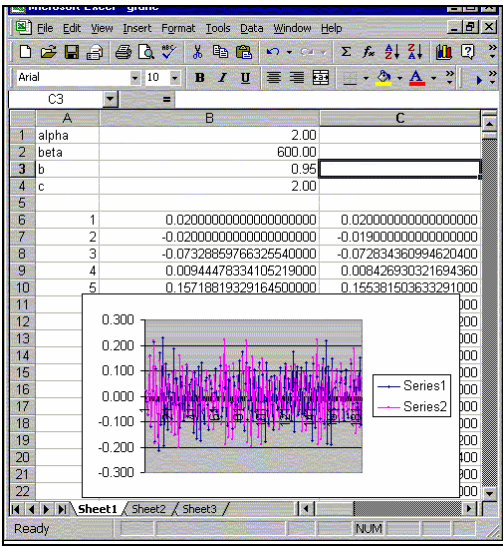


Fig. 1.4. A case in which we have sensitive dependence on the initial conditions

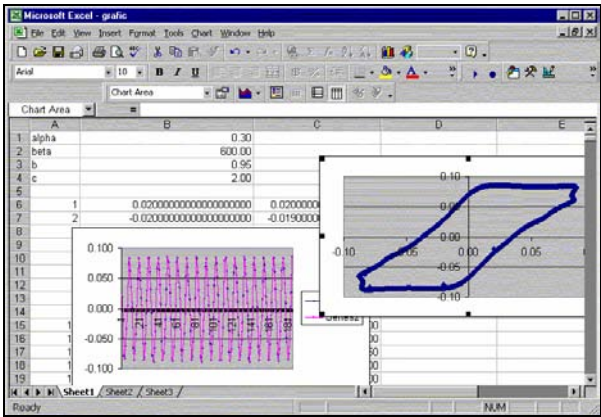


Fig. 1.5. A case in which we do not have sensitive dependence on the initial conditions

1.3.2Implementation method used to calculate the Lyapunov exponents, using Householder QR Factorization

In order to obtain the Lyapunov exponents in the case in which we have a chaotic attractor (the trajectory image suggests a chaotic attractor) or a limit cycle, we cannot use their definitions and for this reason we suggest for

implementation the method proposed in [1], which uses the Householder QR factorization. In this case we must calculate approximations for Lyapunov exponents until t has a high value (generally, we can conclude only after 200 000 -300 000 iterations).

We propose to calculate the Lyapunov exponents approximations for groups of iterations (for example, 6000), and in the same way like in Subsection 1.3.1, we save only the values necessary to continue the algorithm for the following group of iterations. All the values, which are not necessary, will be deleted. In this way, we can quickly obtain a good approximation of the Lyapunov exponents. In the case in which we do not use groups of iterations (independently of the used software application) and we calculate the approximations for a greater number of iterations, the time necessary to obtain the results can be a serious problem.

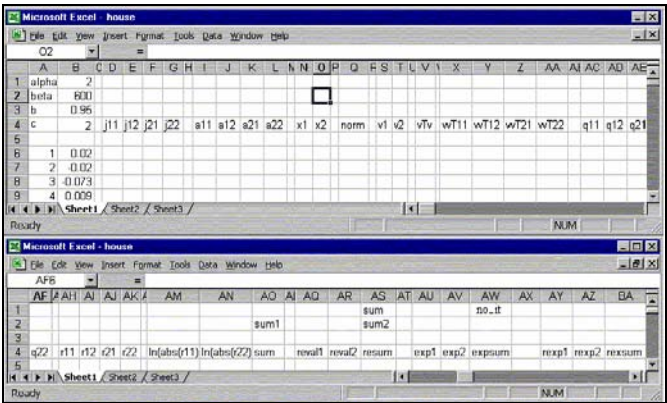


Fig. 1.6. Implementation method for calculate Lyapunov exponents

In order to obtain quickly the results, our implementation method proposes that we follow some steps.

Here we will use the algorithm presented in the Subsection 1.2.2 and the dates presented in Figure 1.6.

Now we present the steps of our implementation:

1. In columns $D:G$ we calculate the elements of matrix J .
2. In columns $I:AO$ we calculate: the elements of matrix A ; the vector x and its norm; the vector v ; the value $v^T v$; the matrices vv^T , Q and R ; natural logarithm of the absolute value of diagonal elements of the matrix R and their sum.
3. We make all these calculations for groups of N iterations (we have choose N to be 6000). At this step, for the two last iterations, we save all the values from the domain from columns $B:AO$ in another domain, like example in columns $CX:EL$.

4. In the domain from columns $AM:AO$ we have formulas. We save all these results as values in the domain from columns $AQ:AS$.
5. We delete all values calculated in the *step 2*.
6. We delete all values calculated in the *step 1*.
7. In the cells $ATI(AUI, respectively AVI)$ we save the sum for the values from columns $AQ(AR, respectively AS)$, corresponding to the current and previous groups of iterations.
8. In the cells AXI we save the value of iterations number for which we have calculated all values in the domain from the columns $B:AS$.
9. The last iterations from a group (see the *step 3*) become the initial condition for the following group of iterations.

We will repeat all the *steps 1-9* for some groups of iterations (e.g., 20-30 groups).

10. Now, we will calculate $N(6000)$ approximations for Lyapunov exponents corresponding to the last group of N iterations. In the domain from the columns $AU:AW$ we have formulas and for a high speed of calculation we make copies of values in the domain from the columns $AY:BA$. In the moment in which we have the copies in the domain from the columns $AY:BA$, we will delete the formulas from the domain from columns $AU:AW$.

If the approximations for Lyapunov exponents corresponding for the last group of N iterations do not lead to a good conclusion on Lyapunov exponents values, we will repeat the *step 9-10* for another number of iterations groups and for the last group we will calculate the new approximations for Lyapunov exponents.

In all our steps, we have used *VBA (Visual Basic for Applications)* macros in *Excel*. The use of macros increases considerably the speed with which we can obtain results. This aspect is very important in order to arrive to a better conclusion regarding our numerical results.

1.4 Numerical simulations

We have made many numerical simulations and we have found many situations in which the system displays a chaotic behavior.

For the particular case where $\alpha=2$, $b=0.95$, $c=2$ and the initial condition $(s_0, s_1) = (0.02, -0.02)$, we investigate the ranges of parameter β for which system (1.2) presents a chaotic or a non chaotic behavior.

We observe different intervals of values for β for which system (1.2), in general, displays a chaotic behavior. These intervals are $[46, \infty)$, $[16, 28]$, $[6.2, 12]$, $[4, 4.6]$, $[2.9, 3.8]$, $[1.7, 1.8]$. Every two intervals presented here are

separated by an interval of values of β which characterizes a sequence of period-doubling bifurcations for system (1.2). We present in Figure 1.7 the strange attractors (in space (s_t, s_{t+1})) which characterize different types of intervals of values for β for which the system displays a chaotic behavior.

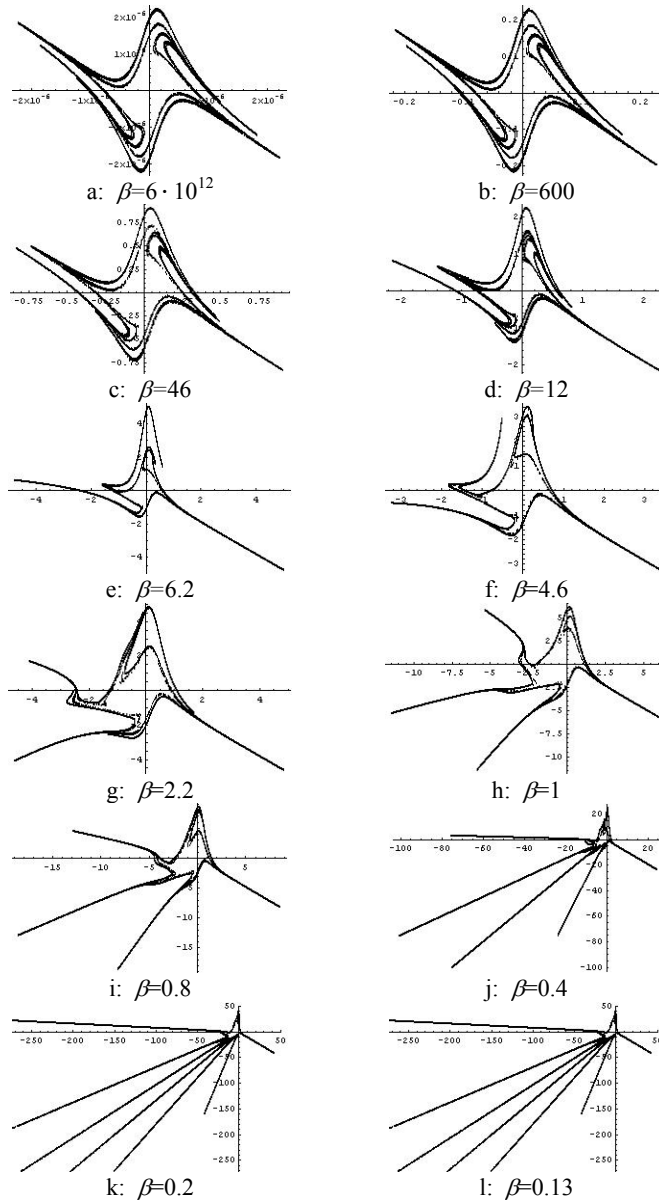


Fig. 1.7. Chaotic attractors in the case $c=2$, $s_0 = 0.02$, $s_1 = -0.02$, $b = 0.95$, $\alpha = 2$

In Table 1.1 we give the values of Lyapunov exponents in the case of the strange attractors present in Figure 1.7.

The images from Figure 1.7 seem to represent the same attractor which increases and is deformed, but probably that is not so clear. The transformation image cannot be obtained statically. In order to obtain an animation we can use different applications (e.g., in *Flash*). Our animation is presented on the page: <http://www.catrinelvoicu.home.ro/chaos.html>.

Table 1.1. Lyapunov exponents in the case $s_0 = 0.02$, $s_1 = -0.02$, $b = 0.95$, $\alpha = 2$

β	λ_1	λ_2
$6 \cdot 10^{12}$	0.3728	-1.2735
600	0.4029	-1.2158
46	0.2685	-1.1734
12	0.3299	-1.341
6.2	0.2636	-1.9012
4.6	0.2244	-1.4617
2.2	0.2033	-1.5614
1	0.09489	-1.424
0.8	0.143	-1.5599
0.4	0.1054	-3.3219
0.2	0.0787	-2.5733
0.13	0.0132	-4.1387

In Table 1.2, we present an example of a sequence of period-doubling bifurcations, displayed by system (1.2). We use the same initial conditions (0.02, -0.02). Here (s_0, s_1) means a periodical point from the periodical cycle towards which tends the trajectory starting from point (0.02, -0.02).

Table 1.2. A sequence of period-doubling bifurcations (the first period is 4)

β	s_0	s_1	Period	λ_1	λ_2
40.5528	0.22816	-0.10586	1024	-0.0012	-0.6448
40.5513	0.49251	0.23373	512	-0.0063	-0.6387
40.52	-0.49253	0.2336	128	-0.0177	-0.6272
40.51	-0.48868	0.228887	128	-0.0486	-0.5963
40.38	-0.49377	0.24238	64	-0.0568	-0.5882
40.3	-0.49496	0.23532	32	-0.0242	-0.6208
39	-0.50179	0.23606	16	-0.0062	-0.6398
36	-0.52696	0.24804	8	0.0920	-0.5538
32	-0.5617	0.26258	4	0.2447	-0.4125
28.3853	-0.58727	0.26821	4	0.0002	-0.6963

1.5 Conclusion

We have presented our implementation methods for detecting the dynamics of a nonlinear system and for the calculation of the Lyapunov exponents. These methods present the way in which we can obtain a very large number of observations with the computer, in a very short time, which, at the same time, is conclusive for the obtained results.

Acknowledgement. Result paper within the research project CNCSIS AT code 67/2007.

References

1. Eckmann J-P, Ruelle D (1985) Ergodic theory of chaos and strange attractors. *Reviews of Modern Physics* 57(3), Part I
2. De Grauwe P, Dewachter H, Embrechts M (1993) *Exchange rate theory: chaotic models of foreign exchange markets*. Blackwell Publishers
3. Juncu VD, Raffei-Naeini M, Dudekb P (2006) Integrated circuit implementation of a compact discrete-time chaos generator analog integrated circuits and signal processing. *Springer Netherlands*, ISSN :0925-1030, 46(3):275-280
4. Ulrike F (2006) Generalized models as a tool to study the stability of nonlinear dynamical systems - Nonlinear dynamics, chaos, and applications. In: 6th *Crimean School and Workshops*
5. Voicu MC (2004) On the computational tools role in the nonlinear dynamics study. In: *Proceedings of ICNPAA 5th*, Timișoara, România, Cambridge Scientific Publishers Ltd, UK, pp 665-672
6. Voicu MC (2004) On the periodical solutions of a nonlinear dynamical system. In: *Proceedings of ICNPAA 5th*, Timișoara, România, Cambridge Scientific Publishers Ltd, UK, pp 657-664
7. http://www.cs.ut.ee/~toomas_1/linalg/lin2/node8.html

Chapter 2

The worm cutter tool profiling based on discreet surfaces representation

Virgil Teodor, Marian Cucu, Nicolae Oancea

Department of Manufacturing Engineering, “Dunărea de Jos” University of Galați, Galați, 111 Domnească Street, zip code 800201, ROMANIA, virgil.teodor@ugal.ro

Abstract. The worm cutter profiling for cylindrical or helical surfaces curl rolling generation represent a problem with the point contact between two surfaces reciprocally enwrapping. In this paper there is proposed an algorithm based on the tangents method and a software, developed in *java* programming language, based on the discreet representation of the surface to be generated. Applications of this algorithm for simple elementary profiles which may be assembled in compound profiles are presented.

Keywords. Surfaces discreet representation, worm cutter profiling, the tangents method

2.1 Introduction

The problem of the worm cutter profiling is a problem of surface enwrapping on second degree — the contact between the surface to be generated, associated with a circular centrod, and the primary peripheral surface of the worm cutter, is a point.

Due to this, the profiling of these tools presumes the use of the intermediary surface method, in two successive steps [2, 3, 4]:

- determining of the rack-gear surface, as a cylindrical surface, reciprocally enveloping with the rack-gear flank, in conditions of linear contact between the two surfaces;
- the rack-gear flank being known, there is determined the primary peripheral surface of the worm cutter as reciprocally enveloping with the rack-gear flank, in the conditions of linear contact between these two surfaces.

So, the study of the reciprocally enwrapping surfaces is reduced to a succession of enwrapping problems with linear contact developed based on Gohman's theorem [2, 3].

For both of these steps, the tangents method [1, 5] may be used for the representation of the profiles in discreet form.

2.2 The rack-gear tool profiling

There are determined the relative movements between spaces associated with the blank centrod and the rack-tool centrod, Figure 2.1

$$\xi = \omega_3^T(\varphi_1) \cdot X - a \quad (2.1)$$

and the reverse

$$X = \omega_3(\varphi_1)[\xi + a], \quad (2.2)$$

where

$$a = \begin{bmatrix} -R_{rp} & -\lambda & 0 \end{bmatrix}^T. \quad (2.3)$$

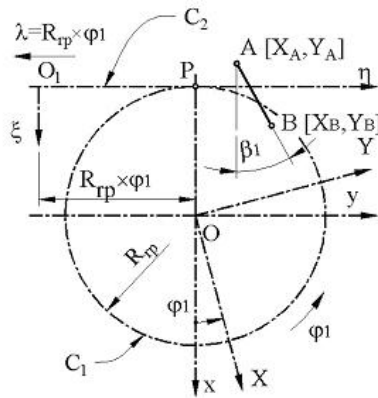


Fig. 2.1. The profile of the curl to be generated

A certain profile, from the curl of the profile to be generated, associated to a circular centre having the radius R_p , may be discretely expressed through the matrix (2.4), referred to the XY system, associated with the whirl of profiles to be generated.

The number of coordinates to define the matrix (2.4) can be correlated to the precision of representing the Σ profile:

$$\Sigma = \begin{bmatrix} X_A & X_2 & \cdots & X_i & \cdots & X_B \\ Y_A & Y_2 & \cdots & Y_i & \cdots & Y_B \end{bmatrix}^T. \quad (2.4)$$

The model used to define the Σ matrix current point coordinates is:

$$\begin{cases} X = X_A + v \cdot \cos \beta_i; \\ Y = Y_A + v \cdot \sin \beta_i, \end{cases} \quad (2.5)$$

where v is an incremental variable.

The Σ family of profiles, described into the $\xi\eta$ reference system, associated with the worked piece, has the form:

$$(\Sigma)_{\varphi_1} \begin{cases} \xi = X_i \cos \varphi_1 - Y_i \sin \varphi_1 + R_p; \\ \eta = X_i \sin \varphi_1 + Y_i \cos \varphi_1 + R_p \varphi_1. \end{cases} \quad (2.6)$$

The enveloping condition, determined to express the generated profile in the form (2.4), by using the tangents method, is

$$\left| \sin \beta_i [Y_i - R_p \sin \varphi_1] - \cos \beta_i [-X_i - R_p \cos \varphi_1] \right| \leq \varepsilon, \quad (2.7)$$

where ε is small enough.

The ensemble formed by the family of profiles of discrete representation (2.6) and the specific enveloping condition (2.7) represents, in a discrete form, the profile of the rack-tool enwrapped to Σ profile in form:

$$I = \begin{bmatrix} \xi_1 & \xi_2 & \cdots & \xi_n \\ \eta_1 & \eta_2 & \cdots & \eta_n \end{bmatrix}^T. \quad (2.8)$$

Based on the algorithm briefly suggested above, an application written in java language was imagined, in order to allow us to find the rack-tool profile by starting from a numerical expression of the profile to be generated.

2.3 Reference systems and generation movements

There are defined the references systems (see Figure 2.2):

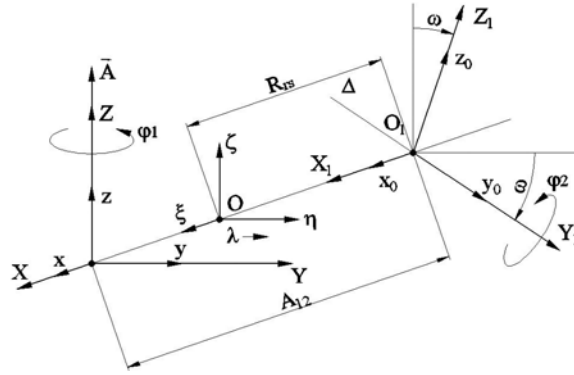


Fig. 2.2. References systems

- xyz is the global reference system, with the z axis superposed on the rotation axis of the orderly surface curl to be generated;
- $x_0 y_0 z_0$ — global reference system, with y_0 axis superposed on the Δ axis, the worm cutter axis;
- XYZ — relative reference system, joined with the orderly surfaces to be generated;
- $X_1 Y_1 Z_1$ — relative reference system, joined with the primary peripheral surface of the worm cutter;
- $\xi \eta \zeta$ — relative reference system, joined with the intermediary surface (the generating rack cutter);
- A_{12} — the distance between axis A (the axis of the orderly surface curl to be generated) and axis Δ (the axis of the worm cutter).

There are defined the correlations between the moving parameters:

- φ_1 is the angular parameter of the rotation around the Z axis;
- φ_2 — the angular parameter of the rotation around the Y_1 axis;
- λ — the translation parameter along the η axis,

$$\lambda = R_{rp} \cdot \varphi_1; \quad (2.9)$$

- R_{rp} — the radius of the centrole associated with the curl to be generated;

$$\lambda = p \cdot \varphi_2 \cdot \cos \omega, \quad (2.10)$$

$$\sin \omega = \frac{R_{rp}}{R_{rs}} \cdot \frac{1}{z}, \quad (2.11)$$

where z is the number of circular pitch of curl to be generated;

- p — the helical parameter of the primary peripheral surface of the worm cutter.

There are known the movements:

$$x = \omega_3^T (\varphi_1) \cdot X, \quad (2.12)$$

which represent the rotation movement of the centrode with the R_{rp} radius;

$$x = \xi + a, \quad (2.13)$$

which represents the $\xi\eta\zeta$ reference system translation, joined with the intermediary surface;

$$x_0 = \omega_2^T (\varphi_2) \cdot X_1 \quad (2.14)$$

representing the rotation of the worm cutter around its own axis;

$$x_0 = \beta(x - B); \quad (2.15)$$

$$B = \begin{Bmatrix} -A_{12} \\ 0 \\ 0 \end{Bmatrix}; \beta = \begin{Bmatrix} 1 & 0 & 0 \\ 0 & \cos \omega & -\sin \omega \\ 0 & \sin \omega & \cos \omega \end{Bmatrix}, \quad (2.16)$$

representing the transformation matrix between the global reference systems.

From (2.12), (2.13), (2.14) and (2.15) the movement assembly is determined by means of the relative movement between relative reference systems $\xi\eta\zeta$ and $X_I Y_I Z_I$:

$$X_1 = \omega_2 (\varphi_2) \beta [\xi + a - B] \quad (2.17)$$

and

$$\xi = \beta^T \omega_2^T (\varphi_2) X_1 + B - a. \quad (2.18)$$

2.4 The primary peripheral surface of the worm cutter

There are defined the principles of the intermediary surface profile determination based on the tangents method.

The intermediary surface may be represented by the matrix:

$$I = \left\{ \begin{bmatrix} \xi_i \\ \eta_i \\ t \cdot j \end{bmatrix}, i = 1, 2, \dots, n \right\}, j = (-m, \dots, -1, 0, 1, \dots, m), \quad (2.19)$$

with the ξ_i, η_i coordinates of the determinate intermediary surface profile, in correlation with the previous methodology;
 t is the linear increment measured along the cylindrical surface generatrix.

2.4.1 The intermediary surface family in the reference system of the worm cutter primary peripheral surface

Based on the equation (2.17) and on the discrete known model of the cylindrical surface, there is determined the cylindrical surface family, in the worm cutter reference system.

After development, the equations are brought to form:

$$(I)_{\varphi_2} \begin{cases} X_1 = [\xi_i + R_{rs}] \cdot \cos \varphi_2 - [\eta_i - \lambda] \cdot \sin \omega \cdot \sin \varphi_2 - t \cdot j \cdot \cos \omega \cdot \sin \varphi_2; \\ Y_1 = [\eta_i - \lambda] \cdot \cos \omega - t \cdot j \cdot \sin \omega; \\ Z_1 = [\xi_i + R_{rs}] \cdot \sin \varphi_2 - [\eta_i - \lambda] \cdot \sin \omega \cdot \cos \varphi_2 - t \cdot j \cdot \cos \omega \cdot \cos \varphi_2; \end{cases} \quad (2.20)$$

with $(i = 1, 2, \dots, n); (j = -m, (-m+1), \dots, -1, 0, 1, 2, \dots, m)$.

The representation of the intermediary surface family in form (2.20) is a points array if a discrete representation of the φ_2 parameter is accepted. Obviously is regarded the link (2.10) between λ and φ_2 is accepted.

The $(I)_{\varphi_2}$ surface family enveloping is the primary peripheral surface of the worm cutter.

2.4.2 The enwrapping condition

There is defined the enwrapping condition in the GOHMAN form,

$$\vec{N}_I \cdot \vec{R}_{\varphi_2} = 0, \quad (2.21)$$

where, from (2.18):

$$\bar{R}_{\varphi_2} = \frac{d\xi}{d\varphi_2} = \beta^T \cdot \dot{\omega}_2^T(\varphi_2) \cdot X_1 - \dot{a}_\lambda \cdot \frac{d\lambda}{d\varphi_2}. \quad (2.22)$$

Finally, the following matrix results:

$$R_{\varphi_2} = \begin{bmatrix} [\eta_i - \lambda] \cdot \sin \omega + t \cdot j \cdot \cos \omega \\ -[\xi_i + R_{rs}] \cdot \sin \omega + p \cdot \cos \omega \\ -[\xi_i + R_{rs}] \cdot \sin \omega \end{bmatrix} \quad (2.23)$$

$$(i = 1, 2, \dots, n); (j = -m, \dots, -1, 0, 1, 2, \dots, m).$$

The normal at intermediary surface is

$$\bar{N}_I = \bar{l}_{i,i+1} \times \bar{k}. \quad (2.24)$$

The assembly of the equations (2.20), (2.21), (2.23) and (2.24) represents the primary peripheral surface in worm cutter form.

2.4.3 The characteristic curve of the conjugated surfaces

The contact curve between the two reciprocally enwrapping surfaces, the I intermediary surface and the S primary peripheral surface of the worm cutter, is defined by the assembly of the equations (2.20), (2.21), (2.23) and (2.24), for $\varphi_2 = 0$.

So, the characteristic curve becomes:

$$C_{I,S} \begin{cases} X_1 = [\xi_i + R_{rs}]; \\ Y_1 = \eta_i \cdot \cos \omega - t \cdot j \cdot \sin \omega; \\ Z_1 = \eta_i \cdot \cos \omega - t \cdot j \cdot \sin \omega, \end{cases} \quad (2.25)$$

with $(i = 1, 2, \dots, n); (j = -m, \dots, -1, 0, 1, 2, \dots, m)$, at which there is added the enwrapping condition (2.21).

The characteristic curve results as the matrix

$$C_{I,S} = \begin{bmatrix} X_{1,i,j} & Y_{1,i,j} & Z_{1,i,j} \end{bmatrix}^T \quad (2.26)$$

$$(i = 1, 2, \dots, n); (j = -m, \dots, -1, 0, 1, 2, \dots, m),$$

for points belonging to I which satisfy the (2.21) enwrapping condition.

2.4.4 Primary peripheral surface of the worm cutter

If the characteristic curve (2.26) has a helical movement around the Y_I axis, with p the helical parameter, a cylindrical helical surface with constant pitch is generated, surface which represents the primary peripheral surface of the worm cutter,

$$S \begin{cases} X_I = X_{I_{i,j}} \cdot \cos v + Z_{I_{i,j}} \cdot \sin v; \\ Y_I = Y_{I_{i,j}} + p \cdot v; \\ Z_I = -X_{I_{i,j}} \cdot \sin v + Z_{I_{i,j}} \cdot \cos v, \end{cases} \quad (2.27)$$

with v as the variable angular parameter.

The axial section of the primary peripheral surface of the worm cutter tool is obtained from equation (2.27) and the condition

$$Z_I = 0. \quad (2.28)$$

The condition to determine the axial section is

$$-X_{I_{i,j}} \cdot \sin v + Z_{I_{i,j}} \cdot \cos v = 0. \quad (2.29)$$

In this way, the assembly of the equations (2.28) and (2.29) represents an in-plane curve — the axial profile of worm cutter, for all of the points which belong to the primary peripheral surface of the worm cutter.

2.5 The software for the Rolling Generating Tool Profiling (RGTP)

The presented algorithm [4, 5] allows the synthesis of the dedicated software, in two successive steps, which allow:

- the rack-gear tool profiling, respectively the I axial section primary peripheral surface profiling reciprocally enwrapping with an ordered surface curl associated with a circular centrode;
- the axial section profiling of the primary peripheral surface of the worm cutter, the S cylindrical helical surface with constant pitch, based on the discreet expression of the I rack-gear flank [2, 6]. There were elaborated the following two flowcharts:
 - the flowchart for the rack-gear profiling reciprocally enwrapping with a Σ profile, represented in a discreet form based on the tan-

gents method, see Figures 2.3, 2.4 software which give the rack-gear crossing section coordinates and graphical form;

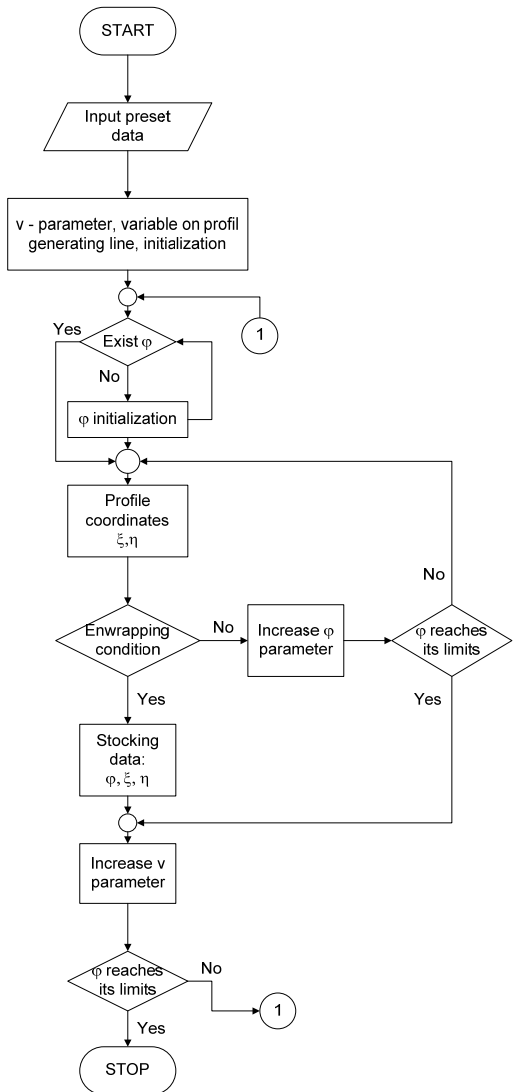


Fig. 2.3. Flow-chart for the rack-cutter profiling

Moreover, in order to ensure the quality check for the proposed method (the tangents method) the software allows, in the first stage, to apply another known enwrapping method to profile the rack-gear reciprocally enveloping with a profile associated with a couple of rolling centrodes.

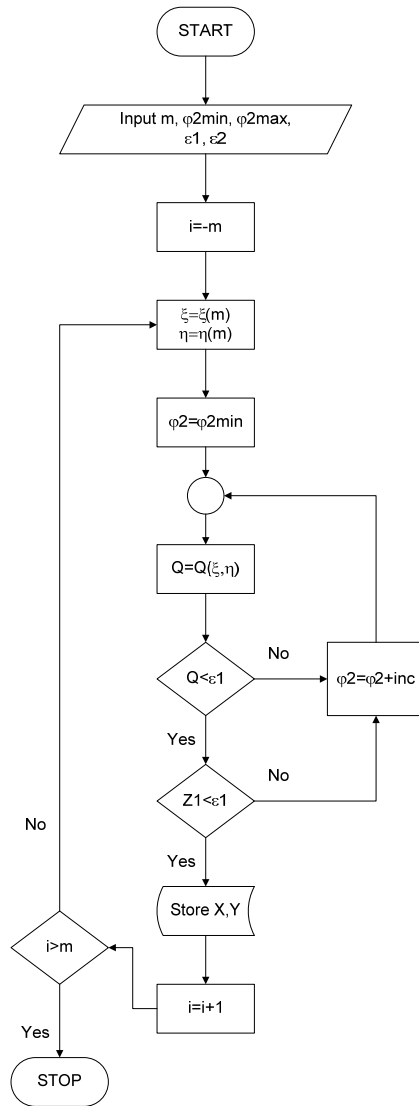


Fig. 2.4. Flow-chart for worm cutter profiling

There was presented the “tangents method” for the discreet representation of surfaces.

For this there was realized java applets, based on the previously presented flow-charts. To access these applets there were made HTML pages which allow the viewing of the specific applets by an internet browser. The structure of these pages and the function of the applets will be presented in the following.

First, the HTML page allows the selection of the machining mode, with a rack-gear, a gear-shaped or a rotary cutter tool.

By accessing the links it is possible to select the profile to be realized, selecting a straight line profile or a circle arc profile regardless of the selected tool.

After the profile to be generated is selected the tool profiling method is chosen.

The navigation is very easy in both senses between all of RGTP software pages, using the internet browser facility. After the selection of the method, the corresponding java applet will be loaded.

In Figure 2.5, there is presented the application in case of a straight line profile generated by a rack-gear tool, profiled with the “tangents method”.

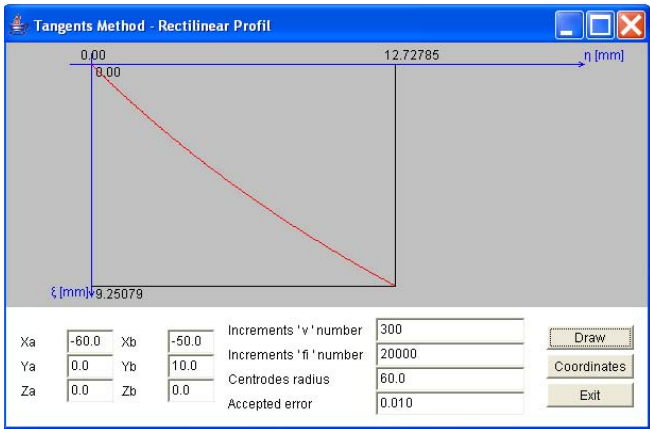


Fig. 2.5. The applet for straight line profile

The presented applets have double use possibilities, as independent applications, as applications which may be viewed with the “Appletviewer” software produced by Sun brand.

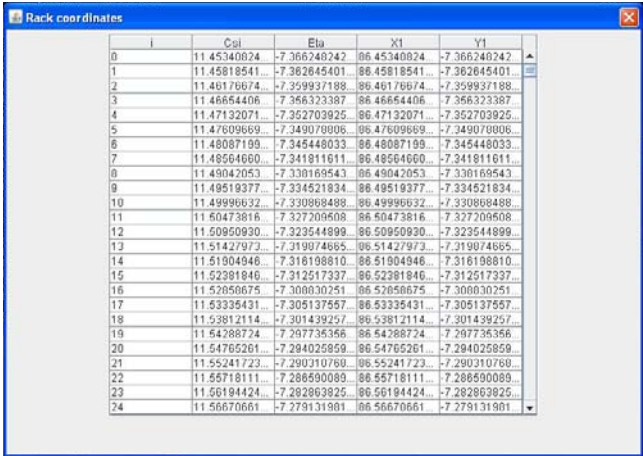
As it is possible to view in Figure 2.5, the use of the applets is very simple and user friendly. The user may modify any of the input data by editing these inside edit box. After clicking on the “Draw” button, the application is launched with new input data and after finishing the calculus a tool graphical representation is made.

By selecting the option “Coordinates”, a table is obtained with all the tool profile calculated coordinates, as it is presented in Figure 2.6.

Because the number of coordinates is very large the viewing window has a scrolling bar. These coordinates may be saved for subsequent using.

The accepted error refers to the enwrapping condition error which is a criterion for tool profile determination.

The graphical representation of the calculated profiles is based on original algorithms. The application determines the maximum scale for the modeled profiles, without distortion on any axis, based on the determination of the display window dimensions. It is considered the obligatory representation of the reference system origin. After the scale determination, it is drawn the profile as a polyline which will include all of the modeled points and the reference system axes are drawn.



i	Cx1	E1a	X1	Y1
0	11.45340024	-7.366240242	86.45340024	-7.366240242
1	11.45818541	-7.362645401	86.45818541	-7.362645401
2	11.46176674	-7.359937188	86.46176674	-7.359937188
3	11.46654406	-7.356323387	86.46654406	-7.356323387
4	11.47132071	-7.352703925	86.47132071	-7.352703925
5	11.47609669	-7.349078006	86.47609669	-7.349078006
6	11.48087199	-7.345448033	86.48087199	-7.345448033
7	11.48564660	-7.341811611	86.48564660	-7.341811611
8	11.49042053	-7.338169543	86.49042053	-7.338169543
9	11.49519377	-7.334521834	86.49519377	-7.334521834
10	11.49996632	-7.330868489	86.49996632	-7.330868489
11	11.50473816	-7.327209608	86.50473816	-7.327209608
12	11.50950930	-7.323544899	86.50950930	-7.323544899
13	11.51427973	-7.319874665	86.51427973	-7.319874665
14	11.51904946	-7.316198810	86.51904946	-7.316198810
15	11.52381846	-7.312517337	86.52381846	-7.312517337
16	11.52858675	-7.308830251	86.52858675	-7.308830251
17	11.53335431	-7.305137557	86.53335431	-7.305137557
18	11.53812114	-7.301439257	86.53812114	-7.301439257
19	11.54288724	-7.297735356	86.54288724	-7.297735356
20	11.54765261	-7.294025859	86.54765261	-7.294025859
21	11.55241723	-7.290310760	86.55241723	-7.290310760
22	11.55718111	-7.286590089	86.55718111	-7.286590089
23	11.56194424	-7.282863925	86.56194424	-7.282863925
24	11.56670661	-7.279131901	86.56670661	-7.279131901

Fig. 2.6. Coordinate viewing

We have to notice that this drawing mode is in the present trend to make web-based interfaces software, very easy to use.

2.6 Applications

Next, it is presented the numerical and graphical results for the worm cutter profiling which is reciprocally enveloping with a circle arc profile.

In Figure 2.7 and Table 2.1, there are shown the form and coordinates of the crossing profile of the rack-gear reciprocally enveloping with the shaft:

$$\begin{aligned}
 X_0 &= -62 \text{ mm}; Y_0 = 0.0 \text{ mm}; Z_0 = 0.0 \text{ mm}; \\
 r &= 10 \text{ mm}; v_{\min} = -60^\circ; v_{\max} = 60^\circ; \\
 z &= 16 \text{ curls}; \sin \omega = \frac{R_{rp}}{z(A_{12} - R_{rp})},
 \end{aligned}$$

see Figures 2.1 and 2.2.

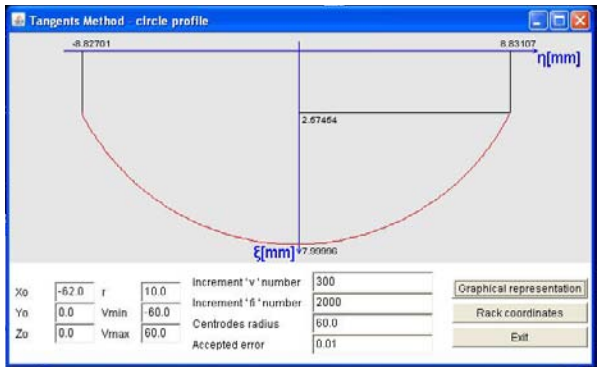


Fig. 2.7. Applet for rack-gear profiling

Table 2.1. Rack-gear profile

Crt. no.	ξ [mm]	η [mm]
0	2.580566	-8.82806
1	2.590754	-8.82297
2	2.600937	-8.81786
⋮	⋮	⋮
998	7.99999	-0.01477
999	8.000001	-0.00367
1000	7.999999	0.006809
⋮	⋮	⋮
1997	2.596056	8.820311
1998	2.585877	8.825407
1999	2.575693	8.830489

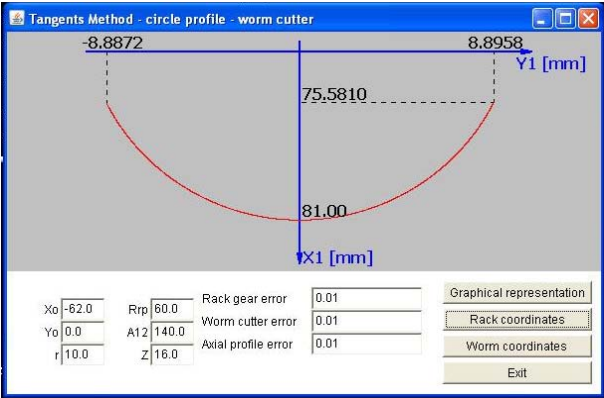


Fig. 2.8. Applet for worm cutter profiling

In Figure 2.8 and Table 2.2, there is shown the axial profile of the worm cutter reciprocally enveloping with the square crossing shaft.

Both profiles are obtained based on original software as application for the presented algorithm.

Table 2.2. Worm cutter profile

Crt. no.	X_1 [mm]	Y_1 [mm]
0	75.58057	-8.82806
1	75.59075	-8.82297
2	75.60094	-8.81786
⋮	⋮	⋮
998	80.99999	-0.01477
999	81	-0.00367
1000	81	0.006809
⋮	⋮	⋮
1997	75.59606	8.820311
1998	75.58588	8.825407
1999	75.57569	8.830489

2.7 Conclusions

The proposed methodology for rack-gear profiling reciprocally enwrapping with an orderly surface curl is based on a particular way to represent the surface—the tangents method.

Based on the intermediary surface specific methodology there were elaborated flow-charts for the two-step worm cutter profiling: the rack-gear profiling (the intermediary surface) and, based on these, the axial section profiling of the worm cutter reciprocally enwrapping with an ordered profile curl.

There have been used the object oriented programming and there has been chosen as programming language the java language, which at this moment is dominant as worldwide programming language for new software. Due to this choice the developed applets have the advantage of being easily understood by other developers and more, it is easy to be reused or subsequently developed due to the inheritance proprieties specific to object oriented programming language.

The software is complex and may select the rack-cutter, gear-shaped cutter or rotary cutter profiling for two elementary profile types: straight line and circle arc profile.

The software allows the comparison of the results obtained by means of the tangents methods with results obtained by others basic or complementary methods universally acknowledged.

References

1. Frumușanu G, Cucu M, Oancea N (2006) Tangents method to profile rack-tool to generate by wrapping. The Annals of "Dunărea de Jos" University of Galați, Fasc V, Technologies in Machine Building, year XXIV (XXIX), ISSN 1221-4566, pp 70-75
2. Litvin FL (1994) Gear geometry and applied theory. Prentice Hall
3. Oancea N (2004) Generarea suprafețelor prin înfășurare. Vol I: Teoreme fundamentale. "Dunărea de Jos" University Foundation Publishing House, ISBN 973-627-106-4, ISBN 973-627-107-2, Galați
4. Oancea N (2004) Generarea suprafețelor prin înfășurare. Vol II: Teoreme complementare. "Dunărea de Jos" University Foundation Publishing House, ISBN 973-627-106-4, ISBN 973-627-170-6, Galați
5. Teodor V, Oancea N, Dima M (2006) Profilarea sculelor prin metode analitice. "Dunărea de Jos" University Foundation Publishing House, ISBN (10)973-627-333-4, ISBN (13)978-973-627-333-9, Galați

Chapter 3

Suggesting correct words algorithms developing in FarsiTeX

Youssef Rezvan,¹ Majid Ghandchi,¹ Farshad Rezvan²

¹Islamic Azad University, Ahar Branch, Dept. Electrical Eng. & Computer Ahar, IRAN, y-rezvan@iau-ahar.ac.ir

²University of Tabriz, Department of Mathematical Sciences, Tabriz, IRAN, frezvan@tabrizu.ac.ir

Abstract. In this paper methods and algorithms of suggesting correct words in a spell checker for Farsi language have been developed, and complexity of them has been calculated. Some software covering these algorithms have been implemented and results of spell checking and receiving suggestions has been discussed. We have concluded the paper by introducing some other techniques.

Keywords. Document Editor, Farsi Language, FarsiTeX, Farsi Spell checking

3.1 Introduction

Although from a long time ago spell checking in English texts are common and people use it easily, for Farsi Language yet, to our knowledge, there does not exists such a suitable system.

Recently some approaches to the design of a Farsi Spell checker for FarsiTeX has been proposed, (Rezvan et al. 2007; Rezvan et al. 2006). FarsiTeX (see Figure 3.1) is an open source typesetting program (Ghodsi 1997) based on LaTeX (Kopka and Daly 1993) version 2.09. LaTeX, a

manuscript preparing program, has been developed over TeX, (Knuth 1986, 1987; Knuth and Mackay 1987).

Here in this paper, we have developed some algorithms for suggesting correct words in a Farsi spell checker. This is mainly in section 3.2 below. Increasing speed of running and controlling memory usages are important issues, which have been covered in sub sequenced sections. We have implemented some spell checking software covering our proposed algorithms to test our discussions. In the conclusion we have also discussed the ways to upgrade this Farsi Spell Checker Version 0.1, as well as other techniques.

When a text is fed to a spell checker, by parsing the text, all of the words of it checked to be in a database of *correct words*. We have such a database of proper spelled words. This database can be increased. We have designed a special facility to do this, which is a program named *dict*. When a text is verified to have correct spells of all its words, it would be suitable to be fed to *dict* program, which adds all its words to the database. This *dict* program has an intelligence of recognizing the fact that "Is a word already in the database?" and if yes, avoiding its repeated addition.

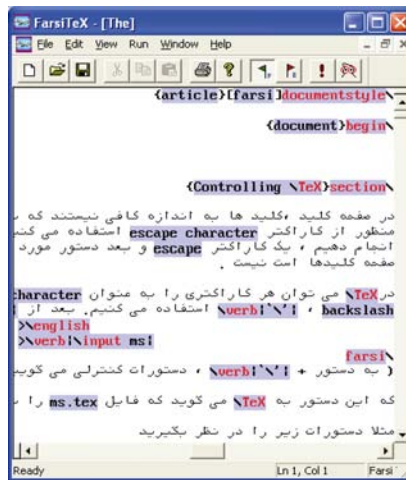


Fig. 3.1. In this version of spell checker we deal just with Farsi words, and ignore any English Letters

To this end, we need a structure to our database. We call this database, a dictionary file. It is obvious that to have a powerful spell checker we need a large dictionary file.

When the spell check program is initiated to check a text, it first loads a dictionary file. If there does not exist any dictionary file, it is better to run first the *dict* program to setup a dictionary file. To this end, we need some

proper spelled texts to cover almost all of the needed words. So to work with this program, at first we need a human spell checking to create a dictionary file. This procedure is normal for any spell-checking program for any language.

3.2 Algorithms of suggesting correct words

The spell checker reads words of the input text one-by-one. This is a procedure performed by a parser and has its own algorithms and problems. On every word, spell checker stops to look it up inside the dictionary. If it is in the dictionary then it is correct, and spell-checker moves to the next word.

If it is a word that isn't inside the dictionary, then it may be an incorrect word.

3.2.1 On an incorrect word

It will be suitable if we reach to a correct word just by minimum changes made on an incorrect word. This is a common facility among spell checkers. To have a good spell checkers it should have some auto corrects or some suggestion correct words that user can select. Methods and algorithms to do this for Farsi language have some similarity to English spell checkers but there are some differences too.

Here we list common algorithms. They all will add words to the suggestion list, if they find a different but similar correct word.

- First we interchange the place of any two adjacent letters. In many cases this procedure finds a correct word, because it is a usual mistake done by typists to press a button before a preceded letter. As an example for a 3-letter incorrect word, this method produces 2 other 3-letter words. If these words are in the dictionary file, they will be placed in the suggestion list.
- Another routine suitable for incorrect words that are typed by missing a letter, inserts all of the 32 letters of Farsi alphabet one by one in the places between each pairs of the letters of the incorrect word and in the two more places one in the beginning of the word and the other at the end of it. for example if a three letter word is incorrect there are four places to add a letter to this word to make it a four letter word so from this three letter word the spell checker produces four times 32 equals 128 new four letter words. All these new words are to be checked to be in the dictionary file, and if it is so, they will be added to the suggestion list.

- The other routine is the dropping routine. This routine drops the letters of the incorrect word, once at a time, and then checks the new words. Again for a 3-letter word, this algorithm finds three new 2-letter words.
- The forth routine is the replacing one. This is suitable for the incorrect word that have a one letter mis-spelled. By this algorithm, the program drops each letter and replace it by all 32 letters of Farsi alphabet, one by one. In this way for a 3-letter incorrect word, we come up with three times 32 equals to 96 new 3-letter words.

We describe some different problems facing in Farsi spell checkers, especially in our Spell Checker.

3.3 Increasing speed

How we can increase the speed of the program run? The speed drawbacks return to the several points including:

- loading the dictionary and,
- searching for every suggested word and other searches.

In spell checker V0.1 searching algorithms are simple Linear Search that searches the whole dictionary every time that they were called. To enhance the process of searching we want to use efficient binary search that specialized to our application. For example using a size definite categorizes. For this we must save our words in objects that can have their own functions such as compare function and to perform searches on the list of these objects.

We should implement the new version of the spell checker in a platform which supports more object orienting methods and has powerful functions in working with strings and lists.

3.4 Memory usage problems

The other problem is the memory usage and now we present some ways to solve them.

3.4.1 Using static functions

At first the words will be saved in objects, which their functions are static ones. Because we have too many objects of this type (Actually every Farsi

word in the dictionary will be an object in the memory) and if each object has non-static function it itself will take place in the memory. So by making them static we create one instance of this type of functions and use this instance with all objects. This is illustrated by Figures 3.2, 3.3.

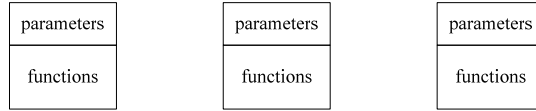


Fig. 3.2. Objects with non-static functions

3.4.2 Using multi files

The other way is to save and load these words in more files. Our suggestion for the method of the splitting is according by what we call weight, and it is related to the applications of the word. More applied words have more weights. At the time when the dictionary is being created, each word is to be counted and if it is used in more places we increase its weight. So we need fair texts to create dictionaries. It means that in our texts to create dictionaries the number of occurrence of a word should be in balance with the normal usage of it in overall texts.

In this way we load at first, words with more weight in the memory, search them and then the other file with lower weight. This novel process cause even faster searches.

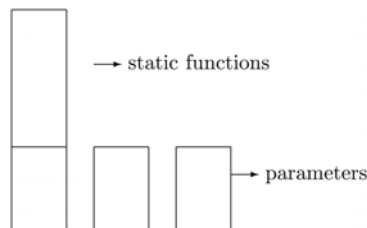


Fig. 3.3. Objects with static functions

Now our dictionary is in several files. At first we load the more weighted file. With this file, we look for the incorrect words in the text. We save these words in a list that we call *mistake list*.

In the second time we load the other file of dictionary that has lower weight. With this file we search just inside the mistake list. If they were founded in this latter dictionary file, we omit them from mistake list. In this way mistake list get shrunk. If now the mistake list were empty, the process has been finished. If not, we should load another file of the dictio-

nary. This process can be continued to find the incorrect words of the input text.

3.4.3 The roots of the Farsi Words instead of the whole words

The other way to decrease the memory usage is to save only the roots of words. This is mentioned in an earlier paper (Rezvan et al. 2007). It means we should have functions to omit the prefix and postfix of a word while creating the dictionary and we should save just the root. Then while finding the mistakes we should have functions to add these postfixes and prefixes. Absolutely it causes the program to run slowly but save a lot of memory.

3.5 Implementation results

Some software has been implemented to test our discussions. In the version 0.1 we ignored any English letters and their spellings. As it is seen in Figure 3.1, a FarsiTeX text may contains many commands written by English alphabet.

During the test we noticed that program has some problems: The first problem is the speed, as in this program we used too many searches so using not so suitable algorithms and saving the words unordered in the file ended in a slow program. So we need to investigate more efficient search algorithms.

The second problem is the using of a lot of space in memory:

1. in the way to completing dictionary, the mass of words that are to be loaded in the memory is growing up. This problem in modern systems with huge amount of memories never seem, but in systems with memories lower than 64MB, it causes more slowly work of the system and forces to load words in the slow virtual memory or the swap memory. We should highlight that this problem has been occurred because of the saving of the all branching of a word in the dictionary file, without any try first to find the roots of the words and save them instead.
2. The other reason is due to the loading of the whole dictionary at once.

These problems discussed in the previous section 3.3 and 3.4 on increasing speed and memory usage problems. We will present features of other versions of the spell checker in another paper.

3.6 Other techniques and conclusion

One of the problems, which are related to the memory in this version 0.1 program, is the data structure of the words that are to be loaded into the memory.

In this program, the data structure that we have used is the C++ vector container that needs a connected memory. To have a better performance, we should get rid of this restriction. We can use better data structures in this program while we will write it with c# platform. These data structures are such as maps, lists and etc, of course for these new data structures we should write its-own search algorithms.

We now describe some implementing techniques that we may use to upgrade the program.

For saving the objects in files we only save the parameters such as string itself. For implementing the binary search algorithm we need to save the words in an ordered way. These causes *make dictionary* application to be more complicated. For search we need a compare function. We put these functions inside objects and we can use more specialized functions for them and even enter the change functions into them too.

The other problem is in the suggestions. These suggestions are not in a good order. By using multi files system for weighting dictionary, some of these problems will be solved. Because it forces to suggest more used words first. For enhancing it at last we need to sort suggestions by their weight using fast sort methods.

The other problem of this program is having not a good interaction with the end user. For solving this problem we propose to use an editor interface and for this we need to have an open source editor, and having a hand over its codes. This new editor may will show mistakes in other colors than the text color or underline them and may suggest words by tool tips. We hope use OpenOffice.Org as our editor.

We developed some algorithms to give hints for the misspelled Farsi words in the texts. We also developed a novel spell checker version 0.1. This can be an acceptable software, but it should be completed to be a user friendly one.

As we saw there are many novel approaches to follow and upgrade the *spell checker* program. These methods are to be tested, measured and to be evaluated and weighted. Benefits of having a good interactive program are seemed to be more than having speedy programs but we hope gradually we find courage to follow the way.

Acknowledgements. We say our best thanks to Dr Jahani, for his encourage and reading the manuscript. The research was supported in part by the Islamic Azad University, Ahar Branch; and in the case of the third author it was also supported by the Department of Mathematical Sciences of university of Tabriz.

References

1. Ghodsi M (1997) FarsiTeX guide (in Farsi Language). Computer department, Sharif University
2. Knuth DE (1986) The TeXBook. Illustrations by Duane Bibby, Reading Mass, Addison-Wesley
3. Knuth DE (1986) TeX: The program. Reading Mass: Addison-Wesley
4. Knuth DE, Mackay P (1986) Mixing right-to-left texts with left-to-right texts. TUGboat 8(1)
5. Kopka H, Daly P (1993) A guide to LaTeX, document preparation for beginners and advanced users. Addison-Wesley
6. Rezvan Y, Ivaz K, Rezvan F (2006) Towards spell checking in FarsiTeX. In: Proc 5th WSEAS Int. Conf. on Data Networks, Communications & Computers, Bucharest, Romania
7. Rezvan Y, Rezvan F, Ghandchi M (2007) FarsiTeX and a novel approach to its spell checker and the dictionary file of it. WSEAS Transactions on Computers 6(2)
8. Rezvan Y, Rezvan F, Ghandchi M (to appear) Evaluation and development of FarsiTeX novel spell checker algorithms.

Appendix

Some view of our program illustrated by figures.



Fig. A.1. This an example of a Farsi text with one spelling error in it



Fig. A.2. This is content of the Errata file, written by our Spell Checker V0.1

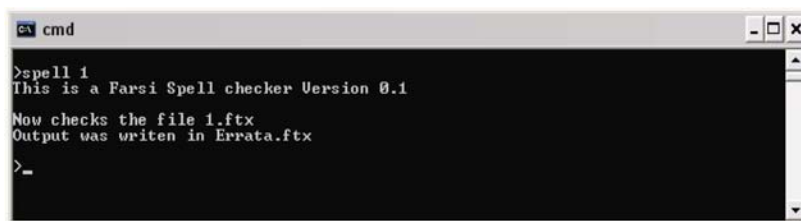


Fig. A.3. We call our Spell Checker in a command prompt as in this figure

Chapter 4

On the analytical and numerical computation in mechanical modeling

Marcel Migdalovici

Romanian Academy, Institute of Solid Mechanics
15, C-tin Mille street, Bucharest, Romania

Abstract. In the paper are studied the possibilities of replacing the manual analytical calculation, that intervene in the mechanical modeling, by an isomorphic numerical calculation which can be performed on digital computers. An algorithm is described for performing the greatest common divisor of two polynomials with several variables that may be used to determine the analytical inverse matrix for a matrix of such polynomials that are used in a mathematical modeling of mechanical phenomena. A new definition of the Euclidean ring is proposed. Is emphasized a question about how much can lead the formal (analytical) calculation in the modeling up to replacing with a numerical method of solution calculation.

Keywords. Mechanical modeling, Shells, Analytical calculation, Coding of operations, Matrix of polynomials

4.1 Introduction about coding

We start the coding of the summing and the product operations in the set of polynomial of several variables with real or integer coefficients with the coding of the algebraic operations for two monomials as follows [11]:

$$C_{a_1 \dots a_m} X_1^{a_1} \dots X_m^{a_m} + C_{b_1 \dots b_m} X_1^{b_1} \dots X_m^{b_m} \rightarrow \{(C_{a_1 \dots a_m}, a_1, \dots, a_m), (C_{b_1 \dots b_m}, b_1, \dots, b_m)\} \quad (4.1)$$

$$C_{a_1 \dots a_m} X_1^{a_1} \dots X_m^{a_m} * C_{b_1 \dots b_m} X_1^{b_1} \dots X_m^{b_m} \rightarrow (C_{a_1 \dots a_m} * C_{b_1 \dots b_m}, a_1 + b_1, \dots, a_m + b_m) \quad (4.2)$$

$$C * C_{a_1 \dots a_m} X_1^{a_1} \dots X_m^{a_m} \rightarrow (C * C_{a_1 \dots a_m}, a_1, \dots, a_m) \quad (4.3)$$

For coding the summing and the product operations of two polynomials we codify the summing and the product polynomials for these polynomials.

In this manner we see that is possible to replace the manual analytical calculation by an isomorphic numerical calculation which can be performed on digital computers.

4.2 Notions on the set of polynomials

A unitary and commutative ring K without divisors of zero is named an integral domain (i.d.).

Let K a factorial ring (f.r.), therefore an integral domain with the property that every non zero and non invertible element of K is a product of prime elements of K [4, 6, 10].

If $a, b \in K$ we say that a divide b if $b = ac$ with $c \in K$. Will be denoted a/b .

A non zero and non invertible element $p \in K$ is named "prime" if for any $a, b \in K$ with p/ab it follows p/a or p/b .

An element $c \in K$ (if exist) is named a greatest common divisor (g.c.d.) of a and b , denoted $c = (a, b)$, if $c/a, c/b$ and if $d/a, d/b$ then d/c .

The elements $a, b \in K$ such that $(a, b) = 1$ are named relatively prime.

Two elements $d_1, d_2 \in K$ such that exists $u \in K$ invertible, with $d_1 = u d_2$ are named adjoints in divisibility.

The ring of polynomials of one variable with coefficients in K is denoted by $K[X]$ and the ring of polynomials of several variables $X_1, \dots, X_n$ with coefficients in K is denoted by $K[X_1, \dots, X_n]$.

Let K i.d. and $f \in K[X]$ of the form

$$f = a_0 + a_1 X + \dots + a_n X^n \quad (4.4)$$

The (g.c.d.) for the coefficients of the polynomial f is denoted by $c(f) \in K$.

If $f \in K[X]$ is of the form (4.4) and $a \in K$ with a/f then a/a_i , where $a_i \in K$, $i = 1, \dots, n$.

If $g \in K[X]$ and $c(g) = 1$ we say that g is primal polynomial.

The ring of polynomials in indeterminate X_m , $m \leq n$, over ring K_0 is denoted $K_0[X_m]$, where $K_0 = K[X_1, \dots, X_{m-1}, X_{m+1}, \dots, X_n]$.

A polynomial $g \in K_0(X_m)$ is of the form

$$g = b_0 + b_1 X_m + \dots + b_n X_m^n \quad (4.5)$$

The coefficients $b_0, b_1, \dots, b_n$ of polynomial g are polynomials of the ring $K[X_1, \dots, X_{m-1}, X_{m+1}, \dots, X_n]$ of polynomials.

If K i.d. then $K[X]$ i.d. and if K f.r. then $K[X]$ f.r..

4.3 A theorem of division

Let K factorial ring, $0 < m \leq n$, with $m, n \in N$ and $K[X_1, \dots, X_n]$ the factorial ring of polynomials with several variables and with coefficients in K .

We formulate bellow the following [9]:

Theorem 1

If polynomials $p_1, p_2 \in K[X_1, \dots, X_n]$, $p_1 \neq 0, p_2 \neq 0$, for fixed m , there exists polynomials $q_1, q_2, r \in K[X_1, \dots, X_n]$, unique without an adjoints in divisibility, such that

$$p_1 q_1 = p_2 q_2 + r \quad (4.6)$$

The polynomial r have the alternative $r = 0$ or $\deg r < \deg p_2$, with degree referred to variable X_m , and the polynomials q_1, q_2, r are relatively prime, $q_1 \neq 0$.

Let now $p_1, p_2 \in K(X_1, \dots, X_n)$, $p_1 \neq 0, p_2 \neq 0$.

The set of divisors with zero remainder for both polynomials p_1 and p_2 is denoted by $D(p_1, p_2)$ and briefly the set of divisors for polynomials p_1 and p_2 is named.

There is the following property [9]:

Theorem 2

In the conditions of first theorem, for fixed m , the equality $D(p_1, p_2) = D(p_2, r)$ is true, where the remainder of the division of the polynomials p_1 and p_2 is r .

This theorem permits to give Euclid's type algorithm for performing the greatest common divisor of two polynomials of several variables with coefficients in factorial ring.

4.4 A new definition of Euclidean ring

N. Jacobson, in his treatise [6], gives the following definition of Euclidean ring:

A domain of integrity D is called Euclidean if there exists a map $\delta: D \rightarrow N$, of D into the set N of non-negative integers, such that if $a, b \neq 0 \in D$ then there exist $q, r \in D$ such that $a = bq + r$ where $\delta(r) < \delta(b)$.

We propose the following definition:

A factorial ring D is called Euclidean if there exists a map $\delta: D \rightarrow N$, of D into the set N of non-negative integers, such that if $a, b \neq 0 \in D$ then there exist $q_1, q_2, r \in D$ such that $a q_1 = b q_2 + r$ where $\delta(r) < \delta(b)$ and q_1, q_2, r are relatively prime.

4.5 The model of thin plates and coding

The thin plate is supposed homogeneous, isotropic and elastic linear. The thickness of the plate is supposed constant and also the hypothesis Love-Kirchoff of the "non-deformed normal element" is considered [3]. We analyze the thin plate in hypothesis that the middle surface is of revolution.

The revolution thin plate is described by the vector of position for the point $P(\theta, z)$ situated on the middle surface:

$$\vec{R}(\theta, z) = r(z)\cos(\theta)\vec{i} + r(z)\sin(\theta)\vec{j} + z\vec{k} \quad (4.7)$$

The vector of displacements with his applied point on the surface is considered of the form:

$$\vec{U} = u\vec{t}_z + v\vec{t}_\theta - w\vec{n} \quad (4.8)$$

In (4.8), $\vec{t}_z$ and $\vec{t}_\theta$ are the unitary vectors attached to coordinate curves concerning the point $P(\theta, z)$.

The first fundamental form of the middle surface is of the form: $ds^2 = r(z)^2 d\theta^2 + (1 + r_z^2) dz^2$ with $r_z = dr(z)/dz$ and the coefficients of the fundamental form $A = r(z)$, $B = (1 + r_z^2)^{1/2}$.

The coefficients of the second fundamental form of the surface are: $D = -r/B$, $D' = 0$, $D'' = r_{zz}/B$, where $r_{zz} = d^2r(z)/dz^2$.

The building of the codified thin plate model on computer is based on coding of the differentiation operation.

For deduction of the codified differentiation of the vector $\vec{U}$, firstly the codified formulas of differentiation for the unitary vectors $\vec{t}_z$, $\vec{t}_\theta$, $\vec{n}$ are deduced.

The vector equilibrium equations of the Goldenveizer model [3], attached to any middle surface point of the plate and to appropriate three dimensional local system of axis, are:

$$-\frac{\partial(r\vec{F}^{(z)})}{\partial z} - \frac{\partial(B\vec{F}^{(0)})}{\partial \theta} + rB\vec{P} = 0. \quad (4.9)$$

$$\begin{aligned} & -\frac{\partial(r\vec{C}^{(z)})}{\partial z} - \frac{\partial(B\vec{C}^{(0)})}{\partial \theta} - rB\vec{F}^{(0)}X\vec{t}_z - \\ & - rB\vec{F}^{(z)}X\vec{t}_\theta + rB\vec{C} = 0 \end{aligned} \quad (4.10)$$

In (4.9) and (4.10), $\vec{P}$ is the external force which action on the plate, $\vec{C}$ is the external resultant moment and

$$\vec{F}^{(z)} = N^{z0}\vec{t}_z - N^z\vec{t}_\theta + Q^z\vec{n}, \quad \vec{C}^z = M^z\vec{t}_z - M^{z0}\vec{t}_\theta, \quad (4.11)$$

$$\vec{F}^{(0)} = -N^0\vec{t}_z - N^{0z}\vec{t}_\theta + Q^0\vec{n}, \quad \vec{C}^0 = -M^{0z}\vec{t}_z - M^0\vec{t}_\theta \quad (4.12)$$

The coding of the operations for the deduction of the model of revolution thin plates, in displacements, is developed in the following fixed order of successive variable which intervenes in the model [11]:

$$E, h, \frac{1}{1+\nu}, \frac{1}{1-\nu^2}, \frac{1}{r}, r, \dots, r_{z5}, \frac{1}{B}, B, u, u_0, u_z, \quad (4.13)$$

$$u_{00}, u_{0z}, u_{zz}, u_{000}, \dots, u_{zzzz}, v, \dots, v_{zzzz}, w, \dots, w_{zzzz}$$

In (4.13) the modulus of elasticity is E , the shell half-thickness is h , Poisson's coefficient is ν and the components of displacement of a point, situated on the middle surface of shell, are u, v, w .

Vector and scalar codified operations of addition, multiplication and differentiation subroutines have been performed.

All the results needed for the mathematical model, expressed in displacements, are deduced on the computer by decoding the analytical expressions and are used as input data for the program of static or dynamic calculus of thin plates [8].

The numerical method used for calculus of the revolution thin plates takes into account the boundary conditions at the extremities $z = z_1$ and $z = z_2$ as well as the development in series of vector displacement components concerning coordinate variables [5, 8].

In this mathematical model the research is compound of two stages, the analytical stage where are defined the equations of the model using the parameters of the physical model and the numerical stage where the values of the parameters are defined and a numerical method of performing the solution is applied.

Many authors (Knuth 1981, Buchberger et al. 1982, Davenport et al. 1987, etc) have analyzed the symbolic computation, but this research is not yet exhausted [1, 2, 7].

4.6 Conclusion

The possibilities of replacing the manual analytical calculation by an isomorphic numerical calculation are investigated.

A question about how much can lead the analytical calculation in the modeling up to replacing with a numerical method of the solutions calculation appeal. The large analytical phase developed in the modeling permit us to increase the precision of the final calculus of solution. Any steps are analyzed here. New strategies in this research are needed, assisted by the computer.

References

1. Buchberger B, Collins GE, Loos R (1982) Symbolic & algebraic computation. Computing Supplementum 4, Springer-Verlag, Wien-New York
2. Davenport J, Siret Y, Tournier E (1987) Formal calculus (in French). Masson, Paris, New York, Barcelone, Milan, Mexico, Sao Paulo
3. Goldenveizer AL (1953) Theory of elastic shells (in Russian). Gostehizdat, Moscow
4. Ion D Ion, Nita C, Nastasescu C (1984) Complements of algèbra (in Romanian). Scientific and Encyclopaedic Publishing House, Bucharest
5. Iovanescu VR (1980) Stress in shells with middle surface of mixed type (in Romanian). Ph.D. thesis, Bucharest University
6. Jacobson N (1973) Basic Algebra vol I. Freeman, San Francisco
7. Knuth DE (1981) The art of Computer Programming vol II Seminumerical Algorithms. Addison-Wesley
8. Migdalovici M (1985) Automation of mechanical structures calculus with application to nuclear power plant (in Romanian). Ph.D. thesis, Bucharest University
9. Migdalovici M (2004) A theorem of division with a remainder in a set of polynomials with several variables. Creative Math. 13:5-10
10. Nastasescu C, Nita C, Vraciu C (1986) Basic Algebra vol I (in Romanian). Publishing House of the Romanian Academy, Bucharest
11. Visarion V, Migdalovici M (1979) On the transposition of the analytical calculation on computers. Rev Roum Sci Techn Mec Appl 24:847-853

Chapter 5

Real genetic algorithm for traveling salesman problem optimization

A.R. Awad,¹ I. Von Poser,² M.T. Aboul-Ela³

¹Department of Environmental Engineering, Tishreen University, Lattakia, Syria

²Ingenieurtechnik, Merck KGaA., Darmstadt, Germany

³Civil Engineering Department, Minia University, Egypt

Abstract. This work aims at solving the Traveling Salesman Problem (TSP) through developing an advanced intelligent technique based on real genetic algorithm (GA). The used GA comprises real-value coding with specific behavior taking each code as it is (whether binary, integer, or real), rank selection, and efficient uniform genetic operators. The results indicated, in comparison with the other applied optimization methods (linear, dynamic, Monte Carlo and heuristic search methods), that the real GA produces significantly the lowest distance (least cost tour) solution. It is concluded that real GA approach is robust and it represents an efficient search method and is easily applied to nonlinear and complex problems of the TSP in the field of solid waste routing system in the large cities.

Keywords. Intelligent technique, real genetic algorithms, optimization, traveling salesman problem, large-scale example

5.1 Introduction

The problem of solid waste and its management is complex in small towns and critical in large metropolitan areas [1]. The routing is one of the main components of solid waste management in the cities where the collection

takes 85% of the solid waste system cost and only 15% for disposal [2]. The traveling salesman problem (TSP) is one of the most widely studied and most often cited problems in operations research. For over fifty years the study of the TSP has led to improve solution methods for wide range of practical problems [3]. Those studies based mainly on several modeling approaches like linear, dynamic programming techniques and heuristic techniques [4, 5, 6]. Genetic algorithms have also been used successfully to solve this NP-problem. However, in many of such works, a little effort was made to handle the nonlinear optimization problem of routing solid waste collection.

In general, the TSP has attracted a great deal of attention because it is simple to state but difficult to solve [7]. The exhaustive algorithms for solving the TSP or node route problem are rarely very good in any sense [8]. They perform well for $n \leq 6$ and very badly for $n \geq 15$; it is time consuming and needs huge storage and memory time. It has been reported by Coney (1988) that a 21-location tour would require 77,100 years of computer time on a million operation per second (PC) [9]. Mathematical programming approaches have had rather limited success with this problem. According to Thieraut and Klekamp (1975) for a 20-node problem, integer linear programming requires 8000 variables and 440 constraints while dynamic programming is limited to 13-node problems [10].

The aim of this paper is to find the optimal routing for solid waste collecting in cities, taking Irbid City in Jordan as an example problem, through developing an advanced intelligent technique based on real genetic algorithm (GA). In addition, a comparison has been made with other optimization techniques such as linear, dynamic, Monte Carlo simulation, heuristic algorithms.

5.2 Encoding schemes of GA

There are many variations of GAs but the following general description encompasses most of the important features. The analogy with nature requires creation within a computer of a set of solutions called a population. Each individual in a population is represented by a set of parameter values that completely propose a solution. These are encoded into chromosomes, which are originally sets of character strings, analogous to the chromosome found in DNA.

The GA search, sometimes with modification, has proved to perform efficiently in a large number of applications. This efficiency lies in the robustness of the search method that underlies the GA approach and in the

flexibility of the formulation itself. In contrast to traditional optimization methods that track only a single pathway to the optimal solution, genetic algorithms cover a whole population of possible solutions. The selection of an appropriate chromosome representation of candidate solutions to the problem at hand is the foundation for the application of genetic algorithms to a specific problem. With a sufficiently large population of chromosomes, adequate representation will be achieved. In real-value coding, variables that can take on continuous values are represented as a real (i.e. continuous) variable in the string. In this case, the string consists of a series of real values.

Real-value chromosomes have been used successfully in multireservoir system operation by various authors [11, 12], and in pipe networks optimization by the authors [13]. In this paper a new GA based methodology for optimal solving the TSP has been applied. This methodology takes by real representation with any code (binary, integer, or real) without any need to transfer from one to another, i.e. our GA, using chromosome as unit, works equally well with integer and non-integer decision variables. This is a specific behavior distinguishes our real-coding GA from other (binary, Gray, or integer) coding GAs. Thus the encoding scheme used herein proved to have the property of matching the effective problem search space towards the GA search space. But here in our example the nodes are taken as integer quantities working with one Chromosome which comprises all nodes encoded as A, B, C and so on (Table 5.1). Nevertheless, this choice does not limit our GA.

Table 5.1. Coding of the network

A	1
B	2
C	3
D	4
E	5
F	6
G	7
H	8
I	9
J	10
K	11
L	12
M	13
N	14
O	15
P	16

← Node No. 16 is added to indicate the ending point.

5.3 Example: A real problem of solid waste routing

The use of GA to solve a traveling salesman problem will be illustrated by using real representation to solve the solid waste network studied initially in Reference [14].

Irbid is divided into six regions which are considered as separate solid waste generation areas. Each of these regions has its own department which regulates the solid waste services in the region. There is no specific routing basis for the vehicles being left to the driver's choice. Occasionally, one pick-up point may be missed. In regions which have two collection vehicles, they may meet at the same pick-up point several times. Once the solid waste is loaded into the vehicles, it is carried out of Irbid to the disposal site located far away.

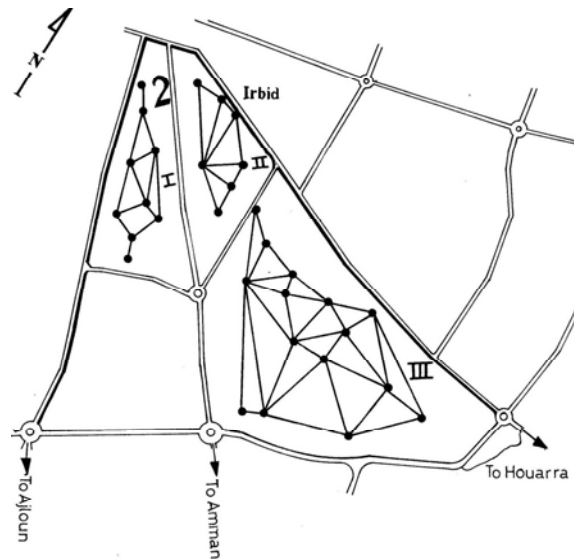


Fig. 5.1. The constructed networks in region two

The suggested procedure for solving the vehicle routing problem in the selected Region 2 of Irbid begins with a particular node closest to the garage or the previous region and ends with the nodes closest to the disposal site. This reduces the number of permutations considerably. Further detailed procedures can be found in Reference [15]. There are thirty-one pick-up points (nodes) in this Region 2 with about fifty containers distributed on it. Two major streets pass through the area of this region and divide the total area into three sectors, each having its own nodes and its own network, and each network has its distance matrix (Figure 5.1). Network I

has 9 nodes (pick-up points), network II has 7 and finally network III has 15 nodes. The network III was used as a case study. A set of 15 nodes with 1 or 2 waste containers at each node were in service by vehicles. The collecting vehicles are equipped with compactors and have to collect the contents of about thirty full containers. The distances between nodes in Network III are shown in Figure 5.2. The routing problem in this part of Irbid is a node routing or traveling salesman problem (TSP).

Our problem has been the construction of a tour through n points with the minimum distance, where the vehicle does not have to return to the starting point, i.e. the number of possible tours is $(n-2)!$. The application based on the fact that the shortest solid waste collection tour should begin with the first node No.1 (closest to the municipality garage), pass by all nodes once and only once, and end with the node No. 15 (closest to the disposal site out of the city).

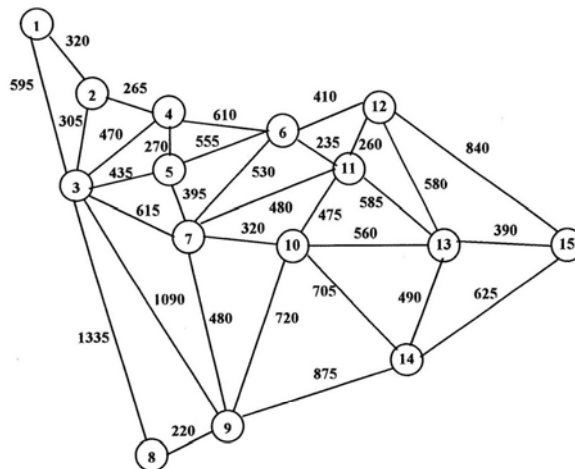


Fig. 5.2. Network (III) of region two

5.4 GA optimization

The following procedures are required for the formulation of GA:

5.4.1 Coding

The genetic algorithm requires that the decision variables describing trial solutions to the solid waste routing problem be represented by a unique

coded string of nodes (pick-up points). This coded string is similar to the structure of a chromosome of genetic code. As for the example, there are 15 decision variables to be made about the network. Each of these decision variables can take one node (pick-up point). A real string made of 15 sub-strings is used for representing the problem into a suitable form for use within a GA (Table 5.1). This string (chromosome) of 15 genes represents a route design for the network consisting of 15 nodes.

It is worthy to mention that the node (No. 16 or P) has not been coded but was used (as pseudo node) to indicate the ending point of the studied network.

5.4.2 Fitness

The fitness of a coded string representing a solution for the traveling salesman problem of solid waste routing is determined by the shortest cycle/cost provided that the collection vehicle passes through all nodes just for once on each.

Table 5.2. Distance matrix among the nodes

	stri ₀	inte ₁	inte ₂	inte ₃	inte ₄	inte ₅	inte ₆	inte ₇	inte ₈	inte ₉	inte ₁₀	inte ₁₁	inte ₁₂	inte ₁₃	inte ₁₄	inte ₁₅	integer ₁₆
stri	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O		
1 A	000	320	595	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000		
2 B	320	10000	305	265	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000		
3 C	595	305	10000	470	435	5000	615	1335	1090	5000	5000	5000	5000	5000	5000		
4 D	5000	265	470	10000	270	610	5000	5000	5000	5000	5000	5000	5000	5000	5000		
5 E	5000	5000	435	270	10000	555	395	5000	5000	5000	5000	5000	5000	5000	5000		
6 F	5000	5000	5000	610	555	10000	530	5000	5000	5000	235	410	5000	5000	5000		
7 G	5000	5000	615	5000	395	530	10000	5000	480	320	480	5000	5000	5000	5000		
8 H	5000	5000	1335	5000	5000	5000	5000	10000	220	5000	5000	5000	5000	5000	5000		
9 I	5000	5000	1090	5000	5000	5000	480	220	10000	720	5000	5000	5000	875	5000		
10 J	5000	5000	5000	5000	5000	5000	320	5000	720	10000	475	5000	560	705	5000		
11 K	5000	5000	5000	5000	5000	235	480	5000	5000	475	10000	260	585	5000	5000		
12 L	5000	5000	5000	5000	5000	410	5000	5000	5000	5000	260	10000	580	5000	5000		
13 M	5000	5000	5000	5000	5000	5000	5000	5000	5000	560	585	580	10000	490	5000		
14 N	5000	5000	5000	5000	5000	5000	5000	5000	875	705	5000	5000	490	10000	5000		
15 O	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	840	390	625	10000		
16 P	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	10		

The evaluation or objective function used is rather simple and determinant of the distance of a routing solution by summing the lengths of the nodes distances making up the network (Table 5.2). The value of 10000m is a hypothetical value points to the node itself, but actually such a case never occurs. In the used coding scheme also infeasible solution can appear. In the distance matrix every infeasible route selection such as “C to J” will be penalized with a high distance of 5000m. An example for fitness evaluation is shown in Figure 5.3.

Infeasible solutions, failed to meet the aforementioned node requirements, are not removed from the population. Instead, they are allowed to join the population and help guide the search, but for a certain price. Although this technique does not exclude infeasible solutions from the population, it reduces the probability of infeasible solutions remaining in future generations and permits convergence from all regions of the parameter space.

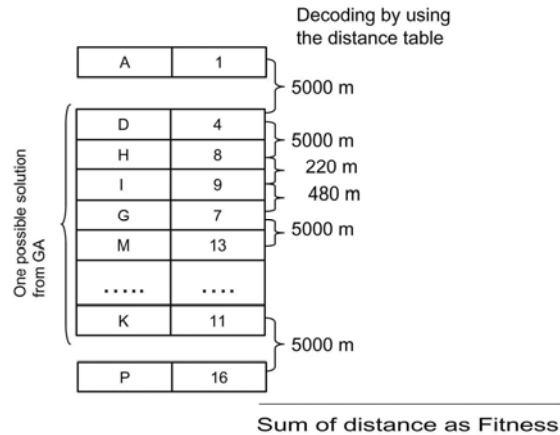


Fig. 5.3. Example for fitness evaluation

5.4.3 Selection, Crossover and Mutation

Detailed explanation and analysis of the selection, crossover and mutation operators for reproduction process using real representation with examples, features and processes can be seen in Reference [14]. The foregoing procedures of the GA formulations are shown in Figure 5.4. The used parameters for implementing our GA technique in the real example are the population size of 40 with probabilities of 0.8 for crossover and 0.4 for mutation. The optimization was carried out over 80 generations.

Furthermore, it is worthy to mention here, that an important aspect of crossover operator in application to a multivariate problem in real-representation comes from the fact that the gene comprises a single allele and it is itself the parameter value; while in the binary or Gray coding the crossover occurs only at the boundaries of the gene which consists of alleles or bits, thus exposing the gene to split. Therefore, what distinguishes our GA work is that the crossover comes only in between the genes, thus avoiding destroying any of them, so the coded information is not de-

stroyed; contrary to what happens with most GAs based on binary coding and others.

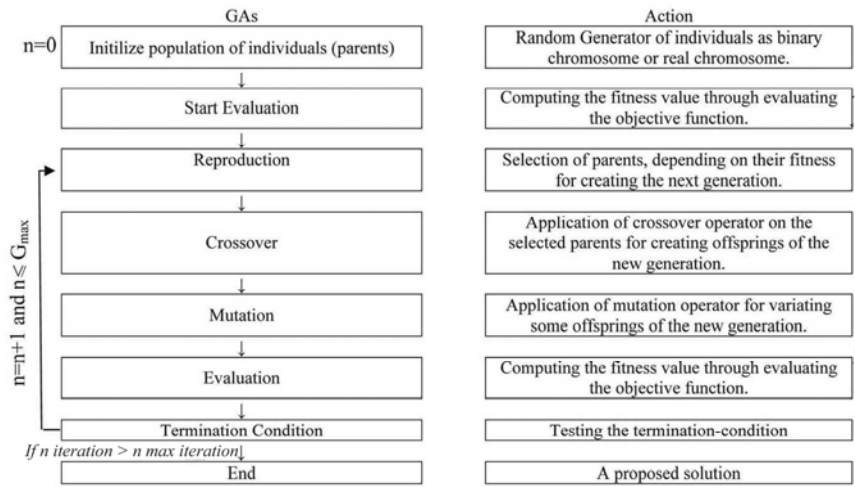


Fig. 5.4. Flow-chart for the GA implementation together with the related actions

5.5 Results

The present work revealed by using linear and dynamic programming techniques, that the mathematical modeling has limited success with the TSP, with limited number of nodes and with the need to a large number of constraints; and that for a case like ours it takes much longer time. These results conform with the other reported results in the literature that mathematical programming approaches have had rather limited success with TSP.

By using GA, a computer program of three operators has been developed. The GA deals with a real-value chromosome comprising the 15 genes representing the nodes (route) network to be optimized. The fitness of the chromosome is computed through the objective or evaluation function, which determines the cost (distance) solution by summing the lengths of the node distances making up the network.

The GA searches for the minimum length of the solid waste routing, thus the objective function is supposed to be minimized. The total search space is 6.2×10^9 possible network routes. The reached results are presented in Figure 5.5, which shows a typical plot of the cost (distance) of the solution in each generation.

These results are presented as worst fitness, average fitness and best fitness to differentiate among them, and to show the optimality of the GA operators program. The GA found the best solution which led to the shortest – distance (lowest cost tour) of 6585m (6595-10) with 2060 simulations (evaluations) for collection vehicles routing in Irbid. Figure 5.6 shows the final route of this best solution. The computing time did not exceed 35 seconds CPU time on a PC (AMD 2.4 GHZ), while in the mathematical methods it takes much longer time and in the heuristic algorithms and their modification and Monte Carlo simulations it takes about 5 minutes.

The quality of the network routes solutions reached and the robustness of the method can be gauged by comparing with the other results from literature presented by the authors [15] who applied their own research work on the same example for modeling techniques of Monte Carlo Simulation and Heuristic Algorithms which resulted that the shortest tour was by Monte Carlo Simulation 6715 m (one million random trials) and by modified heuristic algorithm was 7945m.

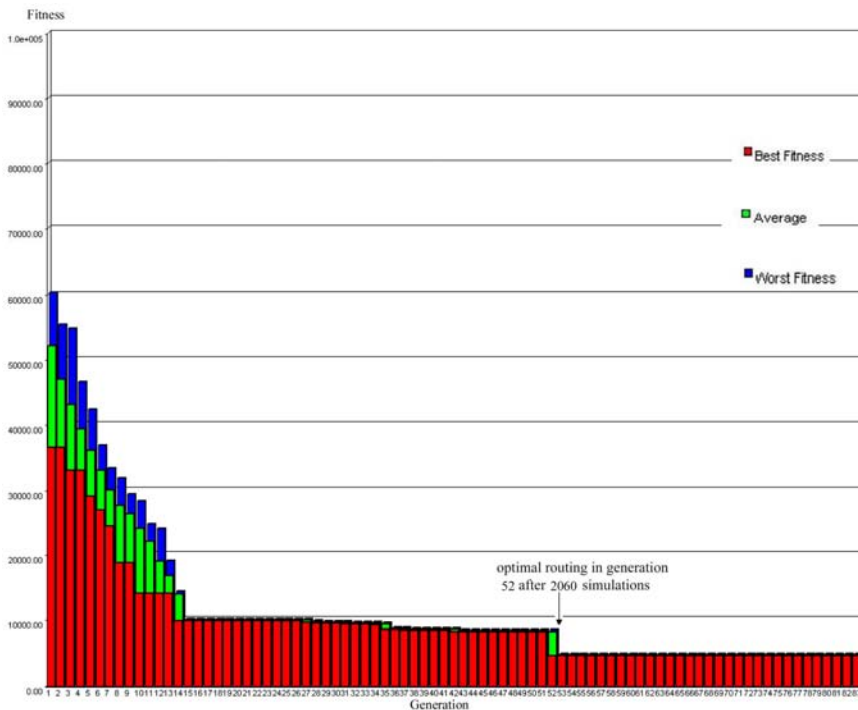


Fig. 5.5. Worst, average and best of generation distances for solid waste routing in the studied network of TSP

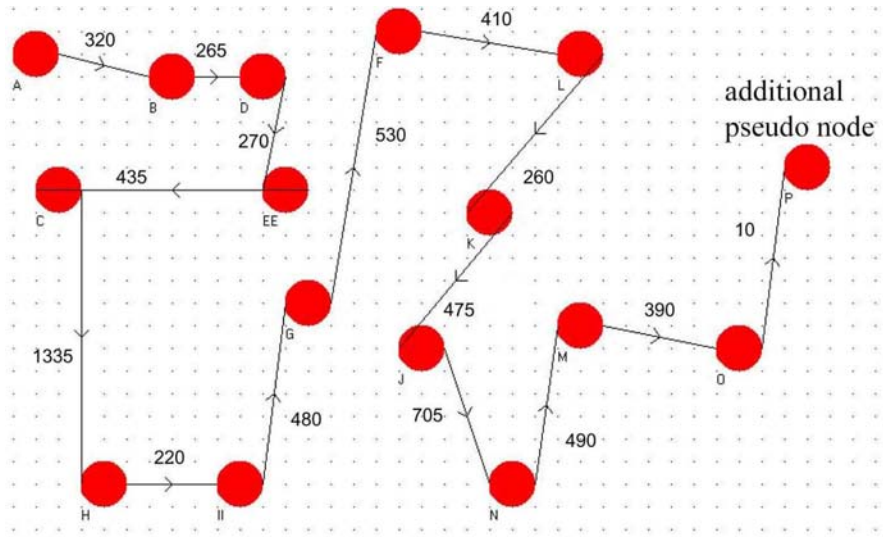


Fig. 5.6. Shortest distance routing for solid waste collection in the studied network of TSP

5.6 Conclusion

The results of our study, in comparison with the other applied optimization methods (linear, dynamic, Monte Carlo and heuristic search method), indicate that the real GA, through its specific behavior and through its efficient operators, presents significantly the lowest distance (cost tour) solution. Accordingly, it is concluded that real GA approach is robust and it represents an efficient search method and is easily applied to nonlinear and complex problems of the TSP in the field of solid waste routing system in the large cities. Furthermore, it has certain advantages over other applied optimization techniques such as linear, dynamic, Monte Carlo simulation and heuristic search methods in both cost and computer time.

Addition of other real constraints such as pick-up points (nodes) or extension of the studied waste collecting network for bigger serviced cities etc., as decision variables can also be incorporated into our real GA by increase in the chromosome length by adding more genes for representing such additions in a route design for waste collecting problem in cities, i.e., more generations would be required to reach the optimum.

It is worthy to remark at the end that our (GA – TSP) program is not complicated to apply, since it doesn't need much mathematical sophistica-

tion for comprehending its mechanisms. So this program can be considered as a tool that gives the designer/decision maker a great number of alternative solutions for traveling salesman problem.

References

1. Koushki AP, Hulsey JL, Bashaw EK (1997) House hold solid waste traits and disposal site selection. *J. Urban Planning & Development*, ASCE, 123(1):1-9
2. Ludwig H, Black R (1968) Report on the solid waste problem. *J. of the Sanitary Engineering Division*, ASCE, 94, SA2:355-370
3. Cook W (2003) The traveling salesman problem and optimization on a grand scale. SIAM Annual Meeting, Montreal
4. Francis R, McGinnis L, White J (1983) Location analysis. *Eur. J. Operational Res.* 12(3):220-252
5. Ramu VN, Kennedy JW (1994) Heuristic algorithm to locate solid waste disposal site. *J. Urban Planning & Development*, ASCE, 120(1):14-21
6. Boyce ED, Lee H, Janson NB, Berka S (1997) Dynamic route choice model of large-scale traffic network. *J. Transportation Eng.*, ASCE, 123(4):276-282
7. Rivett P, Ackoff RA (1963) *Manager's guide to operational research*. New York: John Wiley and Sons Inc.
8. Goodman S, Hedetniemi S (1985) *Introduction to design and analysis of algorithms*. New York: Mc Graw-Hill
9. Coney W (1988) Traveling salesman problem solved with simulation techniques. *Int. J. of Systems Science* 19(10):2115-2122
10. Thieraut R, Klekamp R (1975) *Decision making through operations research*. New York: John Wiley and Sons Inc.
11. Oliveira R, Loucks DP (1997) Operating rules for multireservoir systems. *J. Water Resour. Res.* 33(4):839-852
12. Wardlaw R, Sharif M (1999) Evaluation of genetic algorithms for optimal reservoir system operation. *J. Water Resour. Plng. and Mgmt.*, ASCE, 125(1):25-33
13. Awad A, Von Poser I (2005) Genetic algorithm optimization of water supply networks. *Water Intelligence Online Journal (WIO)*, International Water Association (IWA) Publishing 8(4):1-25
14. Abu-Hassan R (1988) Routing of solid waste collection vehicles in Irbid, M.Sc. Thesis, Department of Civil Engineering, Jordan University of Science and Technology
15. Awad A, Aboul-Ela MT, Abu-Hassan R (2001) Development of a simplified procedure for routing solid waste collection. *International Journal for Science and Technology (Scientia Iranica)* 8(1):71-75

Chapter 6

The COTECOMO: CONstructive Test Effort COst MOdel

Ljubomir Lazić¹, Nikos Mastorakis²

¹School of Electrical Engineering, Vojvode Stepe 283, Belgrade, Serbia,
llazic@vets.edu.yu

²Military Institutions of University Education, Hellenic Naval Academy,
Terma Hatzikyriakou, 18539, Piraeus, Greece, mastor@ieee.org

Abstract. The primary purpose of Software Testing Process and Evaluation (STP&E) is to reduce risk. While there exists extensive literature on software cost estimation techniques, industry practice continues to rely upon standard regression-based algorithms. These software effort models are typically calibrated or tuned to local conditions using local data. This paper cautions that current approaches to model calibration often produce sub-optimal models because of the large variance problem inherent in cost data and by including far more effort multipliers than the data supports. Building optimal models requires that a wider range of models be considered while correctly calibrating these models requires rejection rules that prune variables and records and use multiple criteria for evaluating model performance. This article compares the approaches taken by three (COCOMO II, FP, UCP) widely used models for software cost and schedule estimation to develop COTECOMO (CONstructive Test Effort COst MOdel). It also documents what we call the large variance problem that is a leading cause of cost model brittleness or instability. This paper proposes Software/System Test Point (STP), a new metric for estimating over all software testing process. STP covers so-called black-box testing; an estimate for the test activities, which precede scenarios (threads) testing (white-box testing included), will already have been included in the estimate produced by function point analysis. Software test point is a useful metric for

test managers, which aids in precise estimation of project effort, addresses the interests of metric group, software managers of the software organization who are interested in estimating software test effort.

Keywords. Test metrics, Software test points, Software/System testing, Software test effort, Test management

6.1 Introduction

Software testing is been considered so important that organizations can assign teams only for testing activities. Testing is an important activity to ensure software quality. Tests are usually run several times to certify that code maintenance did not accidentally insert defects into working parts of the software. In such situations, test teams shall be able to estimate the required effort to execute test cases in its schedules and to request more resources or negotiate deadlines when necessary. When regarding model-based testing approaches that we use in our documented Integrated and Optimized Software Testing methodology (IOSTP) [1-4], a high number of test cases can be automatically generated. As team resources are limited, it may be not practical to execute all generated test cases. Their complexity usually determines the effort required to execute them and it can be used for planning test resources and test suites. Several software development estimation models have been proposed over the years. However, these models do not estimate the effort for executing a given test suite, since their estimations are based on software development complexity instead of its test planning, test case design and test execution complexity. According to our reading of the literature (e.g. [5-14]), “best practices” in model-based effort estimation include:

- Local calibration (or LC); i.e. using local data to set two special tuning parameters;
- Stratification; i.e. given a database of past projects, and a current project to be estimated, restrict local calibration to just those records from similar projects.

These two approaches can easily be shown to fall short of what should be expected from a “best practice”. If stratification improved performance, then subsets of the data should usually generate better models with lower error rates than models learned from all the data. Similarly, while LC sometimes produces the best models, often it does not. We compares standard LC with the multiple data mining algorithms in our COTECOMO

(CONstructive Test Effort COst MODEL) which is extended and adapted of T. Menzies' et al. [21] COSECKMO effort estimation workbench applied to IOSTP [1]. COTECOMO is a general workbench containing multiple data mining tools, all of which are unleashed on a data set and a best method is selected on a dataset-by-dataset basis. While standard methods sometimes do well, data mining algorithms yielded models with overall smaller mean error and much smaller standard deviation. The most prominent results from this study are that comparing the performance of development efforts estimators to select a best estimator is problematic [5, 21]. The variance in the performance seen over these small data sets is very large, for example, the standard deviations are often much larger than the mean. Such large deviations complicates comparisons across different data sets. Hence, we must compare model performance using some heuristic rejection rules that compare more than just mean performance.

Note that this paper is about finding better practices for automatic model-based effort estimation. For a discussion on best practices in manual expert-based estimation, see [14]. There are few research articles focusing on test effort estimation model [6,9]. The proposed software test effort estimation metric, Software/System Test Point (STP), has been conceptualized based on the fact of COTECOMO effort estimation procedure and case study results that accuracy of analysis increases when the system components are dealt with separately. Also, as testing plays a significant role in the software engineering community as a software reliability improvement activity and is being considered as a separate branch of software engineering, the need of estimating system test effort separately has been felt. Many formulae proposed for software size estimation [14] have predefined values. The values specified in the formulae vary from project to project and organization to organization. Hence, this paper proposes a framework, which can be adopted by organizations to estimate software/system test effort. The proposed metric is a generalized metric for software/system testing which can be tailored to the operational or architectural processes of the organization or project. Menzies [21] argue that elaborate and expensive testing regimes will not yield much more than inexpensive manual or simple automatic testing schemes. Pol, Veenendaal [9] have developed and applied Test Points successfully to estimate test effort for black-box testing.

This paper proposes a framework for estimating the software/system test effort using a new metric called Software/System Test Points - STP. The organization deploying this metric initially has to conduct a survey to calculate the weights of the identified attributes, which will be used in estimating the software/system test points. Estimated STP should be compared against the actual STP to improve the estimation accuracy for the subse-

quent projects in adaptive manner. The metric proposed is suitable for the projects having a software/system testing team headed by a test manager or project manager. In scenarios, where the system testing team is an independent organization, it is assumed that the necessary information/data for estimating STP is provided to the test manager.

This paper starts with a brief introduction of the attributes identified for the proposed metric, which affect system test effort. The survey conducted for identifying the relationship between the 12 identified attributes on the system test effort has been explained. Simulation has been performed to calculate the correlation coefficients of the 12 identified attributes. The proposed methodology for estimating STP has been explained together with the COTECOMO methodology of evaluating STP by applying it for two projects. The two projects considered for estimating and evaluating STP have independent testing teams headed by a test manager. The two projects were operating at the same level of maturity.

This paper is organized as follows. Section 6.2 presents the Attributes Affecting Software/System Test Effort. Section 6.3 introduces the COTECOMO methodology context and new STP estimation method. The COTECOMO effort estimation procedure and case study are presented in Section 6.4. Finally, the paper is concluded in Section 6.5.

6.2 Attributes affecting software/system test Effort

Test Engineering covers a large gamut of activities to ensure that the final product achieves some quality goal. These activities must be planned well in advance to ensure that these objectives are met. Plans are based on estimations. In the early years, the Waterfall model has been applied to software development. This model looks upon test engineering as merely a stage in the entire development lifecycle. When techniques evolved over the years for estimating development time and effort, the concept of estimating test-engineering time was overlooked completely. Test engineering is seldom planned for in most organizations and as a result, products enter the market insufficiently tested. Negative customer reactions and damage to the corporate image is the natural consequence. To avoid this, the correct development lifecycle must be chosen and planning should be done early on in the cycle as in our IOSTP [1-4]. Estimation is basically a four-step approach:

1. Estimate the size of the development product. This is either in LOC (Lines of Code) [10-14] or FP (Function Points) [8] or concept of using UCP (Use Case Points) [6,7,22] and our STP estimation model.

2. Estimate the effort in person-months or person-hours.
3. Estimate the schedule in calendar months.
4. Estimate the cost in currency.
5. Find best estimator from these test efforts estimation models by applying COTECOMO estimation rules.

6.2.1 Conventional approach to test effort estimation

Test engineering managers use many different methods to estimate and schedule their test engineering efforts. Different organizations use different methods depending on the type of projects, the inherent risks in the project, the technologies involved etc. Most of the time, test effort estimations are clubbed with the development estimates and no separate figures are available. Here is a description of some conventional methods in use:

1. Ad-hoc method

The test efforts are not based on any definitive timeframe. The efforts continue until some pre-decided timeline set by managerial or marketing personnel is reached. Alternatively, it is done until the budgeted finances run out.

This is a practice prevalent in extremely immature organizations and has error margins of over 100% at times (see Figure 6.1).

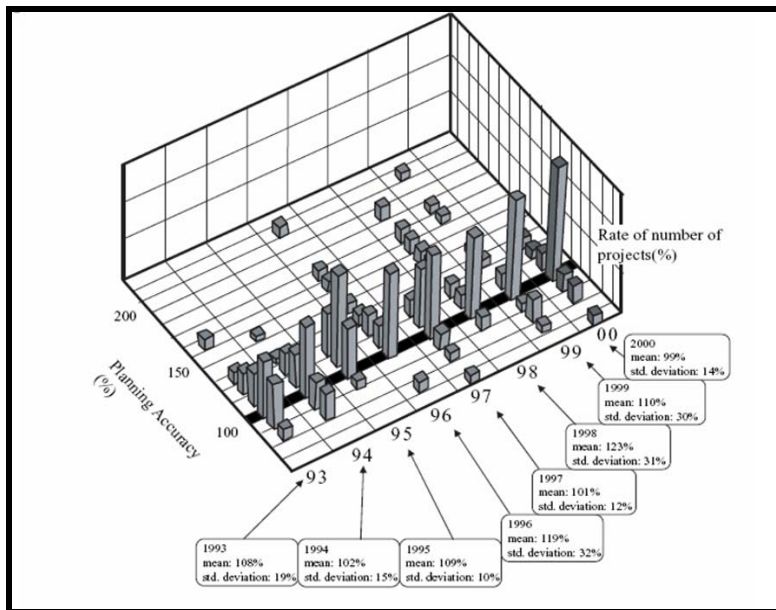


Fig. 6.1. Trend of cost estimation accuracy [5]

2. Percentage of development time

The fundamental premise here is that test engineering efforts are dependent on the development time / effort. First, development effort is estimated using some techniques such as LOC or Function Points. The next step is using some heuristic to peg a value next to it. This varies widely and is usually based on previous experiences. This method is not defensible since it is not based on any scientific principles or techniques. Schedule overruns could range from 50 – 75% of estimated time. This method is also by far the most used.

3. From the Function Point estimates

Capers Jones [5] estimates that the number of test cases can be determined by the function points estimate for the corresponding effort. The formula is Number of Test Cases = (Function Points)^{1.2}. The actual effort in person-hours is then calculated with a conversion factor obtained from previous project data. The disadvantage of using FP is that they require detailed requirements in advance. Another issue is that modern object-oriented systems are designed with Use Cases in mind and this technique is incompatible with them [6, 7, 22].

6.2.2 The selection of the sizing and estimation methodologies

The selection of the Sizing and Estimation methodologies and the tailoring is a task of management knowing that:

- Ideal process models suitable for each environment do not exist
- Process models must be tailored to the respective organization and project situation
- The assigned process model must be enhanced and permanently adapted to the changed basic conditions

Sizing and estimation methodologies

Sizing and estimation methodologies are used for identification and planning of effort and time schedule of software projects, more precisely, just of total effort and total time schedule. They were developed mainly from the analysis of finished projects, using stochastic and/or heuristic technologies. This applies at least to the statistical/ mathematical methods. Other methods however formalize an analogy approach, the derivation from characteristic numbers or use simply a regression. The diversity of the approaches nearly knows no limits and one could mine, it is as numerous as the planning methods and programming languages themselves.

The estimation methodologies determine environmental factors (cost drivers), which could have a positive or negative impact to the project. Depending on the estimation method there could be just one single factor e.g. productivity up to unlimited factors, which even must be defined by yourself. As result of the estimation you get quasi correction factors for the values of a software project. If these factors are applied to the base value from the sizing method, you get a good (real numbers) estimate of the total effort and the total time schedule.

Basic considerations to estimates and efforts influencing areas (see Figure 6.2):

- Estimates are not absolute numbers and have always deviations
- The deviations can be calculated for each method, whereby each method possesses other deviations according to its model
- The deviation of an estimate can be very high depending upon implementation within or outside of its basic condition (specification, extensions, environmental factors/cost drivers etc.)
- The values for the deviation depend on the regarded method and the correct computation (average value simply relative error or squares of error etc.)
- A comparison require always resembles basic conditions

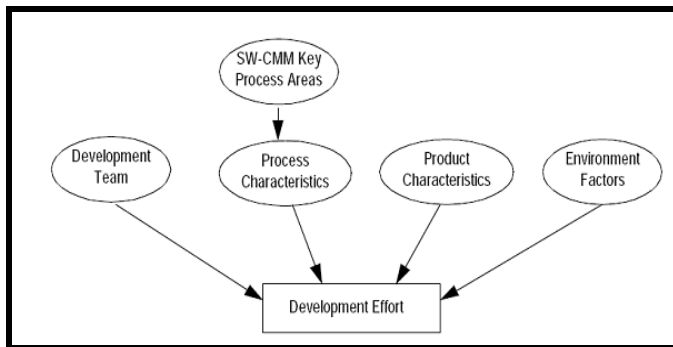


Fig. 6.2. Efforts influencing areas

Characteristics of the cost estimation

- The effort which have to be spend determines the costs of the project
- The determination of the environmental factors (cost drivers) is a major task of the cost analysis

- As basis of the computations and for comparison and/or determination of the environmental factors (cost drivers) different metrics/methods exists

General advantages of estimation methodologies:

- Substantial potential to:
 - Reduce the effort
 - Optimization of time schedule of software projects
 - Improvement of Quality even in field of frequently changed degree of characteristics and not standardized specifications
- Much more accurate than expert judgment (in relation of 25% : 75%)
- Deliver reproducible results
- Supply real numbers instead of random results
- Form the basis for a toolbox
- Support of numerous departments of an enterprise
- Show the real equivalent of a project (target costs)
- Create transparency
- Protect the trust ship between client and vendor by supplying a traceable and reproducible estimation
- The application of estimation methodologies is a critical success factor for software projects.

Many factors affect effort of testing, the accuracy of estimate increases when all the factors are considered.

Broadly, system test effort depends upon the process of testing, type of testing, and test tools used in the organization. The 12 attributes identified for the proposed metric that affect system test effort are as follows:

1. Observed requirement stability
2. Observed software stability
3. Test process tools used
4. Testing tools used
5. Type of testing (e.g., Black box, White Box, etc.) to be performed
6. Test artifacts required
7. System knowledge of the tester
8. Complexity of the system
9. Availability of reusable test cases and test artifacts
10. Environment (UNIX, Windows, Web based, Client-Server)
11. Application type (Real time systems, Database applications)
12. Required software reliability

The descriptions of the attributes identified are as follows:

6.2.3 The software test effort models - critical review

Each of the different “schools of thought” has their own view of the prediction problem despite the interactions and subtle overlaps between process and product identified there. Furthermore each of these views model a *part* of the problem rather than the whole. Perhaps the most critical issue in any scientific endeavor is agreement on the constituent elements or variables of the problem under study. Models are developed to represent the salient features of the problem in a systemic fashion. This is as much the case in physical sciences as social sciences. Economists could not predict the behavior of an economy without an integrated, complex, macroeconomic model of all of the known, pertinent variables. Excluding key variables such as *savings rate* or *productivity* would make the whole exercise invalid. By taking the wider view we can construct a more accurate picture and explain supposedly puzzling and contradictory results. Our analysis of the studies shows how empirical results about test effort and defect density can make sense if we look for alternative explanations. Collecting data from case studies and subjecting it to isolated analysis is not enough because statistics on its own does not provide scientific explanations. We need compelling and sophisticated theories that have the power to explain the empirical observations. The isolated pursuit of these single issue perspectives on the quality prediction problem is, in the longer-term, fruitless. Part of the solution to many of the difficulties presented above is to develop prediction models that *unify* the key elements from the diverse software quality prediction models. We need models that predict software quality by taking into account information from the development process, problem complexity, defect detection process, and design complexity. We must understand the cause and effect relations between important variables in order to explain why certain design processes are more successful than others in terms of the products they produce. It seems that successful engineers already operate in a way that tacitly acknowledges these cause-effect relations. After all if they didn’t how else could they control and deliver quality products? Project managers make decisions about software quality using best guesses; it seems to us that will always be the case and the best that researchers can do is:

1. Recognize this fact and
2. Improve the “guessing” process.

We, therefore, need to model the subjectivity and uncertainty that is pervasive in software development. Likewise, the challenge for researchers is in transforming this uncertain knowledge, which is already evident in elements of the various quality models already discussed, into a prediction model that other engineers can learn from and apply. Ultimately, this re-

search is aiming to produce a method for the statistical process control (SPC) of software production implied by the SEI's Capability Maturity Model.

Effort estimation is a challenge in every software project. The estimates will impact costs and expectations on schedule, functionality and quality. While expert estimates are widely used, they are difficult to analyze and the estimation quality depends on the experience of experts from similar projects. Alternatively, ore formal estimation models can be used. Traditionally, software size estimated in the number of Source Lines of Code (SLOC), Function Points (FP) and Object Points (OP) are used as input to these models, e.g. COCOMO and COCOMO II [12].

Because of difficulties in estimating SLOC, FP or OP, and because modern systems are often developed using the Unified Modeling Language (UML), UML-based software sizing approaches are proposed. Examples are effort estimation methods based on use cases [6,22] and software size estimation in terms of FP from various UML diagrams [17,22]. The *Use Case Points* (UCP) estimation method introduced in 1993 by Karner estimates effort in person-hours based on use cases that mainly specify functional requirements of a system [15]. Use cases are assumed to be developed from scratch, be sufficiently detailed and typically have less than 10-12 transactions. The method has earlier been used in several industrial software development projects (small projects compared to this study) and in student projects. There have been promising results and the method was more accurate than expert estimates in industrial trials [18,19]. An early project estimate helps managers, developers, and testers plan for the resources a project requires. As the case studies indicate, the UCP method can produce an early estimate within 20 percent of the actual effort, and often, closer to the actual effort than experts and other estimation methodologies [20]. Moreover, in many traditional estimation methods, influential technical and environmental factors are not given enough consideration. The UCP method quantifies these subjective factors into equation variables that can be tweaked over time to produce more precise estimates. Finally, the UCP method is versatile and extensible to a variety of development and testing projects. It is easy to learn and quick to apply. The author encourages more projects to use the UCP method to help produce software on time and under budget.

Industry Case Studies

From the time Karner produced his initial report in 1993, many case studies have been accomplished that validate the reasonableness of the UCP method. In the first case study in 2001, Suresh Nageswaran published the

results of a UCP estimation effort for a product support Web site belonging to large North American software company [6]. Nageswaran, however, extended the UCP equation to include testing and project management coefficients to derive a more accurate estimate. While testing and project management might be considered non-functional requirements, nevertheless they can significantly increase the length of the project. Testing a Java 2 Enterprise Edition implementation, for example, may take longer than testing a Component Object Model component; it is not unusual to spend significant time coordinating, tracking, and reporting project status. *Nageswaran's extensions to the UCP equation produced an estimate of 367 man-days, a deviation of 6 percent of the actual effort of 390 man-days.*

In a recent research his author, Nageswaran said he had also applied the UCP method to performance testing, unit-level testing, and white box testing. In the second case study, research scientist Dr. Bente Anda [18] evaluated the UCP method in case studies from several companies and student projects from the Norwegian University of Science and Technology that varied across application domains, development tools, and team size. *For the above studies, the average UCP estimate is 19 percent; the average expert estimate is 20 percent.* Additionally, at the 2005 International Conference on Software Engineering, Anda, et al. [19] presented a paper that described the UCP estimate of an incremental, large-scale development project that was within 17 percent of the actual effort. In the third case study, Agilis Solutions and FPT Software partnered to produce an estimation method, based on the UCP method that produces very accurate estimates. In an article that was presented at the 2005 Object-Oriented, Programming, Systems, Languages, and Applications conference, Edward R. Carroll of Agilis Solutions stated:

After applying the process across hundreds of sizable (60 man-months average) software projects, we have demonstrated metrics that prove an estimating accuracy of less than 9 percent deviation from actual to estimated cost on 95 percent of our projects. Our process and this success factor are documented over a period of five years, and across more than 200 projects [20]. *To achieve greater accuracy, the Agilis Solutions/FPT Software estimation method includes a risk coefficient with the UCP equation.*

6.2.4 Adaptive Cyclic Pairwise Rejection rules of Estimation model candidates using performance measures (ACPREPM)

We consider COCOMO II, FP, UCP and STP estimation model candidates using standard performance measures as many authors do like Menzies [21] and Nagappan [16]. The rejection rules are the core of COTECOMO

effort estimation workbench. In our own work, we revised the COSEEKMO effort estimation workbench for assessing different estimation models based upon COCOMO data sets using data mining techniques despite large deviations rules from [21, 16] until they satisfied certain sanity checks. Large deviations in effort estimation area require model performance comparison using some heuristic rejection rules that compare more than just mean performance data. The results for each treatment were compared using each treatment's of MMRE, SD, PRED(N) and correlation. MMRE comes from the magnitude of the relative error, or MRE, the absolute value of the relative error:

$$MRE = |predicted - actual| \quad (6.1)$$

The mean magnitude of the relative error, or MMRE, is the average percentage of the absolute values of the relative errors over an entire data set. Given T tests (estimators), the MMRE is:

$$MMRE = \frac{100}{T} \sum_i^T \frac{|predicted_i - actual_i|}{actual_i} \quad (6.2)$$

The standard deviation, or SD, is root mean square of predicted and actual effort deviation. Given a T tests (estimators), the SD is:

$$SD = \frac{100}{T} \sum_i^T (predicted_i - actual_i)^{1/2} \quad (6.3)$$

The PRED(N) reports the average percentage of estimates that were within N% of the actual effort candidates' estimated values. Given T test examples, then:

$$PRED(N) = \frac{100}{T} \sum_i^T \begin{cases} 1 \cdot if \cdot MRE_i \leq \frac{N}{100} \\ 0 \cdot otherwise \end{cases} \quad (6.4)$$

For example, a PRED(20)=50% means that half the estimates are within 20% of the actual effort data.

Another performance measure of a model predicting numeric values is the correlation between predicted and actual values. Correlation ranges from +1 to -1 and a correlation of +1 means that there is a perfect positive linear relationship between variables.

We choose estimating model candidates, according to our reading of the literature (e.g. [5-22]), only those which satisfy starting criteria (SC) defined by:

$$\text{MMRE} \leq 10\% \text{ and } \text{SD} \leq 10\% \quad (6.5)$$

than we apply adaptive cyclic pairwise rejection of estimation model in our COTECOMO framework. This stratification of estimator candidates improve performance, then this estimator subsets of the estimators performance data [16, 21] should usually generate better models with lower error rates than models learned from all the collected estimators performance data.

Adaptive Cyclic Pairwise Rejection of Estimation model

The treatments were examined in pairs and if either seemed to perform worse, that one was rejected. This process repeated until no treatment could be shown to be worse than any other. The remaining treatments were called the survivors and were printed.

In addition to these sanity checks, the following model evaluation rules were used in this study:

- *Rule 1* is a statistical test condoned by standard statistical textbooks. If a two-tailed t-test reports that the mean of two treatments is statistically different, then we can check if one mean is less than the other.
- *Rule 2* (which checks for correlation) is a common check used at JPL: the best effort model tracks well between predicted and actual values. Without rule 2, many parts of our data produce multiple survivors.
- *Rule 3* is added in case there is more than one model with the same average performance and R^2 . The model with the smallest MMRE is selected using equation (2)
- *Rule 4* was added because $\text{PRED}(20)=50\%$ using equation (4) is our required performance measure for effort models.
- *Rule 5* This rule rejects treatments that have similar performance, but use more attributes.

Different rules are applied because they measure different aspects of model performance. Figure 6.3 shows adaptive cyclic pairwise rejection rules algorithm using standard statistical tests on MMRE, where an **error** is MMRE, **worse** function that apply **statistically Different** criteria comparing two MMREs x and y , using, well known from statistics, a two-tailed t- test at the 95% confidence interval, i.e.

$$\frac{|mean(x) - mean(y)|}{\sqrt{(sd(x)^2/(n(x)-1)) + sd(y)^2/(n(y)-1))}} > 1.96$$


```

function worse(x,y)
  if statisticallyDifferent(x,y)
  then
    if error(x) < error(y)      then return y fi # rule1
    if error(y) < error(x)      then return x fi # rule1
  else
    if correlation(x) < correlation(y) then return x fi # rule2
    if correlation(y) < correlation(x) then return y fi # rule2

    if sd(x)/mean(x) < sd(y)/mean(y) then return y fi # rule3
    if sd(y)/mean(y) < sd(x)/mean(x) then return x fi # rule3

    if pred(x) < pred(y)          then return x fi # rule4
    if pred(y) < pred(x)          then return y fi # rule4

    if |Subset(x)| < |Subset(y)|  then return y fi # rule5
    if |Subset(y)| < |Subset(x)|  then return x fi # rule5
  fi
  return 0 # if no reason to return true

```

Fig. 6.3. Adaptive Cyclic Pairwise Rejection rules algorithm

6.3 COTECOMO methodology

We propose software test estimation “*The Integrated Methodology driven Estimation points out opportunities to optimize our business processes*” approach, over the our IOSTP [1-4], so called CONstructiveTestEffort-COstModel - COTECOMO methodology that provide a fully customized and tailored solution to perfectly match software testing needs. In some ways, End-to-End (E2E) Architecture Testing is essentially a "gray box" approach to testing -- a combination of the strengths of white box and black box testing. In white box testing, a tester has access to, and knowledge of, the underlying system components. Although white box testing can provide very detailed and valuable results, it falls short in detecting many integration and system performance issues. In contrast, black box testing assumes little or no knowledge of the internal workings of the system, but instead focuses on the end-user experience -- ensuring the user is getting the right results in a timely manner. Black box tests cannot typically pinpoint the cause of problems. Nor can they ensure that any particular piece of code has been executed, runs efficiently, and does not contain memory leaks or other similar problems. By merging white and black box testing techniques, End-To-End Architecture Testing eliminates the weaknesses inherent in each, while capitalizing on their respective advantages. This paper cautions that cur-

rent approaches to model calibration often produce sub-optimal models because of the large variance problem inherent in cost data and by including far more effort multipliers than the data supports. Building optimal models requires that a wider range of models be considered while correctly calibrating these models requires rejection rules that prune variables and records and use multiple criteria for evaluating model performance. This article compares the approaches taken by three (COCOMO II, FP, UCP) widely used models for software cost and schedule estimation to develop COTECOMO (CONstractive Test Effort COst MODEL). It also documents what we call the large variance problem that is a leading cause of cost model brittleness or instability. We applied the E2E Test strategy in our Integrated and Optimized Software Testing framework (IOSTP). In determining the best source of data to support analyses, IOSTP with embedded RBOSTP considers credibility and cost of each test scenario i.e. concept. Resources for simulations and software test events are weighed against desired confidence levels and the limitations of both the resources and the analysis methods. The program manager works with the test engineers to use IOSTP with embedded RBOSTP [3] to develop a comprehensive evaluation strategy that uses data from the most cost-effective sources; this may be a combination of archived, simulation, and software test event data, each one contributing to addressing the issues for which it is best suited.

The central elements of IOSTP with embedded RBOSTP are: the acquisition of information that is credible; avoiding duplication throughout the life cycle; and the reuse of data, tools, and information. The system/software under test is described by objectives, parameters i.e. factors (business requirements - NR are indexed by j) in requirement specification matrix, where the major capabilities of subsystems being tested are documented and represent an independent i.e. input variable to optimization model. Information is sought under a number of test conditions or scenarios. Information may be gathered through feasible series of experiments (E): software test method, field test, through simulation, or through a combination, which represent test scenario indexed by i i.e. sequence of test events. Objectives or parameters may vary in importance α_j or severity of defect impacts. Each M&S or test option may have k models/tests called modes, at different level of credibility or probability to detect failure β_{ijk} and provide a different level of computed test event information benefit B_{ijkl} of experimental option for cell (i,j) , mode k , and indexed option l for each feasible experiment depending on the nature of the method and structure of the test. Test event benefit B_{ijkl} of feasible experiment can be simple ROI or design parameter solution or both etc. The cost C_{ijkl} , of each experimental option corresponding to (i,j,k,l) combination must be estimated through standard cost analysis techniques and models. For every feasible

experiment option, tester should estimate time duration T_{ijkl} of experiment preparation end execution. The testers of each event, through historical experience and statistical calculations define the E_{ijkl} 's (binary variable 0 or 1) that identify options. The following objective function is structured to maximize benefits and investment in the most important test parameters and in the most credible options. The model maintains a budget, schedule and meets certain selection requirements and restrictions to provide feasible answers through maximization of benefit index - $B_{enefit}I_{ndex}$:

$$B_{enefit}I_{ndex} = \max_{i,j,k,l} \sum_j \sum_i \sum_k \sum_l \alpha_j \beta_{ijk} B_{ijkl} E_{ijkl} \quad (6.6)$$

Subject to:

$$\sum_j \sum_i \sum_k \sum_l C_{ijkl} E_{ijkl} \leq BUDGET \quad (\text{Budget constraint});$$

$$\sum_j \sum_i \sum_k \sum_l T_{ijkl} E_{ijkl} \leq TIMESCHEDULE \quad (\text{Time-schedule constraint})$$

$$\sum_l E_{ijkl} \leq 1 \quad \text{for all } i,j,k \quad (\text{at most one option selected per cell } i, j, k \text{ mode})$$

$$\sum_k \sum_l E_{ijkl} \geq 1 \quad \text{for all } i,j \quad (\text{at least one experiment option per cell } i, j)$$

6.3.1 Test effort estimation using system test points: a new metric for practitioners

The ideal system test process considered for conceptualizing the software/System Test Point (STP) metric which quantify effort for the various activities to be performed for software/system testing. The activities of system testers start from the requirements phase. It is a best practice to review the requirements document, which helps in addressing the non-testable requirements during the upstream software development. The preparation and review of test artifacts can be performed once the requirements are baselined. The preparation of test cases and its peer review can be performed during the design phase. Some of the test teams perform the review of design document as well. The test cases prepared will be executed during the integration and system testing phase. The phases, activities and artifacts, might vary from project to project and organization to organization. The process group or software manager needs to identify the relevant activities that will be performed in their project depending upon the project commitments. Many factors affect effort of testing, the accuracy of estimate increases when all the factors are considered. Our metrics group collects the data pertaining to projects such as system test effort, to-

tal effort of the project, size of the software and rating for 12 identified attributes, and calculates the correlation of the 12 identified attributes with respect to system test effort. The identified the 12 attributes (designated with TP) for the proposed metric that affect system test effort are as follows (also shown in Table 1):

1. Observed requirement stability
2. Observed software stability
3. Test process tools used
4. Testing tools used
5. Type of testing (e.g., Black box, White Box, etc.) to be performed
6. Test artifacts required
7. System knowledge of the tester
8. Complexity of the system
9. Availability of reusable test cases and test artifacts
10. Environment (UNIX, Windows, Web based, Client-Server)
11. Application type (Real time systems, Database applications)
12. Required software reliability

In this paper, the top-down approach was followed for calculating the weights used in system test point estimation. A survey has been conducted for calculating the effect of 12 identified attributes on system testing effort. Eleven respondents were identified based upon their experience in software management, specifically in test management and testing. The experience of the respondents varied from 1-15 years. The mean and mode of the respondents experience was 5 years. Respondents have been provided with a questionnaire. The questionnaire used for the survey had the 12 attributes and brief description of the attributes to enhance understanding of the respondents. Weighting represents the influence of the attribute on system test effort. The weights have been calculated using the respondents rating. Data collected from respondents has been simulated using the RISK 4.0 software. Montecarlo simulation has been performed to calculate the correlation coefficients. Correlation coefficients of simulated data have been considered as weights of the attributes. The results represent that the identified attributes have a positive correlation on the system test effort with the magnitudes as shown in Table 6.1. Results show that the requirements stability and type of testing has a minimum effect when compared with the artifacts and reliability on system test effort. The other factors that have higher affect on system test effort include reusability of test cases, application type and usage of testing process tools. Weights in Table 6.1 might vary from organization to organization, due to the varied project settings. So, either the organization can have its own values or the values need to be calculated based on the data collected across the organizations.

Broadly, system test effort depends upon the process of testing, type of testing, and test tools used in the organization. The cognitive factors and skills are not considered for calculating STP. An STP estimation results for our case study of two projects #1 and 2 is shown in Table 6.1. The TP factors affecting the software/system testing have weights (designated with W_i in equation 6.7) that correspond to correlation coefficients and the manager has provided corresponding rating (designated with R_i in equation 6.7) as per the instructions for rating the attributes. The STP method has been applied to a Project#1 and Project#2 and it was observed that the values obtained from estimation were approximately the same as the actual results. Test managers provided ratings for Project#1 and Project#2 as shown in Table 1. Subsequently, the system test points and percentage of system test effort have been estimated. The total size of the project in function points is considered for calculating percentage of system test effort. Systems test points have been estimated using equation:

$$\text{Estimated System Test Points (ESTP)} = C_{adj} * NR * (\sum W_i * R_i) \quad (6.7)$$

Where $C_{adj} = 6.74 \cdot 10^{-5}$, is particular adjustment constant, which value is set in order to make **ESTP** and **FP** values to be comparable and expressed in percentage. Number of business requirements is denoted with **NR**. Percentage of System Test effort has been estimated using equation:

$$\text{Percentage of Estimated System Test Effort (PESTE)} = (\text{ESTP} / \text{FP}) * 100 \quad (6.8)$$

The results of the estimation for Project#1 and Project#2 are also shown in Table 6.1. System test effort can be estimated based upon the percentage of system test effort and the project effort.

6.3.2 Software/System Test Point (STP) evaluation

STP has been evaluated for two projects, Project#1 and Project#2. The actual system test points were calculated, based on the actual software/system test effort, but this data we will have when project ended. The actual system test effort is calculated using formulae:

$$\begin{aligned} \text{Actual Software/System Test Effort (in person-months)} = & \text{Requirements review effort of test team} + \\ & \text{Design review effort of test team} + \text{Test artifact development effort} \\ & + \text{Test artifact review effort} + \text{Manual testing} + \\ & \text{Automated testing} + \text{Performance testing.} \end{aligned}$$

Table 6.1. STP applied for Project#1 and Project#2

Test factor	Description	Weights	Rating Proj. #1	Rating Proj. #2	Proj. #1 Wi*Ri	Proj. #2 Wi*Ri
TP1	Requirements stability	6	8	2	49.32096	12.33024
TP2	Software stability	10	6	1	61.72826	10.28804
TP3	Testing process tools	12	3	8	35.60357	94.94285
TP4	Testing tools	10	6	8	61.26588	81.68784
TP5	Type of testing	7	8	4	55.79434	27.89717
TP6	Artifacts	14	8	6	112.2052	84.15389
TP7	System knowledge of the tester	9	4	4	35.60357	35.60357
TP8	Complexity	8	8	9	63.19248	71.09154
TP9	Reusable test cases	12	7	6	87.12085	74.67502
TP10	Environment	10	4	2	38.06962	19.03481
TP11	Application	12	4	4	48.85858	48.85858
TP12	Reliability	13	4	4	52.8659	52.8659
NR					460	340
ESTP					21.74	14.05
FP					75	52
Ratio	ESTP/FP=				0.289867	0.270192
AverageRatio =						0.280

The actual percentage of Software/System Test Effort is calculated using formula:

$$\text{Percentage Actual Software/System Test Effort (PASTE)} = (\text{Actual system test effort} / \text{Total effort of the project}) * 100 \quad (6.9)$$

The actual system test points have been calculated using formula:

$$\text{Actual Software/System Test Points (ASTP)} = (\text{Actual system test effort} / \text{Total effort of the project}) * \text{Function points} \quad (6.10)$$

The actual software/system test effort and software/system test points for Project#1 and Project#2 are listed in Table 6.2.

The variation in the estimated and actual values was calculated. The percentage of effort spent for software/system testing is approximately the same as the estimated percentage of system test effort for the projects. The difference (PASTE-PESTE) between percentage actual system test effort and percentage estimated system test effort is represented by 'δ'. The difference (ASTP-ESTP) between actual system test points and estimated system test points is represented by 'Δ'. The 'δ' and 'Δ' values are calculated for projects #1 and #2, and the results are shown in Table 6.3.

Table 6.2. Actual values of Project#1 and Project#2

Actual test effort data	Proj. #1	Proj. #2
Actual software/system test efforts (person-months)	7	24
Percentage of Actual software/system test efforts (PASTE)	22.58	32
Actual software/system test points (ASTP)	18.06	16.64

Table 6.3. Variation of actual and estimated values

	PASTE	PESTE	δ	ASTP	ESTP	Δ
Project#1	22.58	27.17	-4.58	18.06	21.738	-3.67
Project#2	32	27.02	4.98	16.64	14.05	2.59

The variation in the actual and estimated STP and percentage of test effort in the estimation accuracy i.e. ‘ δ ’ and ‘ Δ ’ is less than 5%, easy to learn and quick to apply. The managers may need to update the weights only when the actual system test points consistently vary from the estimated system test points. There, also, have been promising results and the method was more accurate than expert estimates in industrial trials [18, 19].

6.4 COTECOMO effort estimation procedure and case study results

Effort estimation is a challenge in every software project. The estimates will impact costs and expectations on schedule, functionality and quality. While expert estimates are widely used, they are difficult to analyze and the estimation quality depends on the experience of experts from similar projects. Alternatively, more formal estimation models can be used. Traditionally, software size estimated in the number of Source Lines of Code (SLOC), Function Points (FP) and Object Points (OP) are used as input to these models, e.g. COCOMO and COCOMO II. Because of difficulties in estimating SLOC, FP or OP, and because modern systems are often developed using the Unified Modeling Language (UML), UML-based software sizing approaches are proposed. Examples are effort estimation methods based on Use Cases Points and software size estimation in terms of FP from various UML diagrams. All these estimation models are candidates in 3 steps COTECOMO effort estimation procedure we defined as follows:

Step 1. As a team (development and test), apply during planning phase:

- 1.1 The Brooks rule using current time expended on the project assuming we are half-way through the "design" phase i.e. 1/2 of total project effort in design phase ("My rule of thumb is 1/3 of the schedule for design, 1/6 for coding, 1/4 for component testing, and 1/4 for system testing.") and 1/2 for test phase.

or

- 1.2 Apply own company rule for planned activities during project realization, similar to nominal percentage given in Table 6.4 from [23].

Step 2. Apply Adaptive Cyclic Pairwise Rejection Rules of Estimation model candidates using performance measures (ACPREPM) to these estimation model candidates:

- 2.1 Compute SLOC by sizing components individually and Effort using COCOMO-II
 - (a) As a developer individually compute SLOC and Effort
 - (b) As a manager estimate SLOC and Effort
 - (c) Find Low, High, and Average of (a) & (b) estimated effort
- 2.2 Apply OO Function Point analysis to estimate efforts and then calculate SLOC/FP ratio for the implementation language used:
 - 2.2.1 Compute Function Point estimations of SLOC as described in D. Garmus and D. Herron's book [8].
 - 2.2.2 Compute SLOC by module breakdown analysis as described in article [7] to:
 - (a) Count Classes to Develop/Modify – Boundary, Entity, Control
 - (b) Size each Class in (a) by counting methods to develop/modify
 - (c) Size each method (using high, medium, low) and assign SLOC count to each size category.
 - (d) Convert SLOC to hours, and hours to work weeks using your performance data.
 - 2.2.3 Find Low, High, and Average of 2.2.1 & 2.2.2 estimated effort
 - Apply the OO Function Point Analysis to Completed Code.
- 2.3 Apply UCP estimation method as described in articles [6, 22].
- 2.4 Apply STP estimation method as described in this article (section 6.3).

Step 3. Compute average estimate of 2.1 – 2.4 data.

Finally, use the STP, FP and UCP methodology on your delivered implementation to arrive at actual STP, FP and UCP and then compute the

SLOC/FP, ESTP/FP and FP/UCP ratio for the implementation language you used.

Table 6.4. Typical software development activities - six types of application [23]

Activities Performed	Web	MIS	Outsource	Commercial	System	Military
01 Requirements	5.00%	7.50%	9.00%	4.00%	4.00%	7.00%
02 Prototyping	10.00%	2.00%	2.50%	1.00%	2.00%	2.00%
03 Architecture		0.50%	1.00%	2.00%	1.50%	1.00%
04 Project plans		1.00%	1.50%	1.00%	2.00%	1.00%
05 Initial design		8.00%	7.00%	6.00%	7.00%	6.00%
06 Detail design		7.00%	8.00%	5.00%	6.00%	7.00%
07 Design reviews			0.50%	1.50%	2.50%	1.00%
08 Coding	30.00%	20.00%	16.00%	23.00%	20.00%	16.00%
09 Reuse acquisition	5.00%		2.00%	2.00%	2.00%	2.00%
10 Package purchase		1.00%	1.00%		1.00%	1.00%
11 Code inspections				1.50%	1.50%	1.00%
12 Independent verification and validation						1.00%
13 Configuration management		3.00%	3.00%	1.00%	1.00%	1.50%
14 Formal integration		2.00%	2.00%	1.50%	2.00%	1.50%
15 User documentation	10.00%	7.00%	9.00%	12.00%	10.00%	10.00%
16 Unit testing	30.00%	4.00%	3.50%	2.50%	5.00%	3.00%
17 Function testing		6.00%	5.00%	6.00%	5.00%	5.00%
18 Integration testing		5.00%	5.00%	4.00%	5.00%	5.00%
19 System testing		7.00%	5.00%	7.00%	5.00%	6.00%
20 Field testing				6.00%	1.50%	3.00%
21 Acceptance testing		5.00%	3.00%		1.00%	3.00%
22 Independent testing						1.00%
23 Quality assurance			1.00%	2.00%	2.00%	1.00%
24 Installation/training		2.00%	3.00%		1.00%	1.00%
25 Project management	10.00%	12.00%	12.00%	11.00%	12.00%	13.00%
Total	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
Activities	7	18	21	20	23	25

In our case study from Project#1 and Project#2, where Java implementation language is used, COTECOMO workbench estimated data are summarized in Table 6.5:

Table 6.5. Estimated and actual values

Estimated and Actual test effort data	Proj. #1	Proj. #2
ESTP	21.74	14.05
FP	75	52
UCP	10	7
Actual SLOC	3750	2600

From Table 6.5 we computed:
 1 FP = 50 Java SLOC
 ESTP/FP = 0.28
 FP/UCP = 7.514

These formulas are very useful to compare various estimation methods.

6.5 Conclusion and future work

An early project estimate helps managers, developers, and testers plan for the resources a project requires. As the case studies indicate, the our STP and UCP method can produce an early estimate within 20 percent of the actual effort, and often, closer to the actual effort than experts and other estimation methodologies [22, 23]. Building optimal models requires that a wider range of models be considered while correctly calibrating these models requires rejection rules that prune variables and records and use multiple criteria for evaluating model performance. This article compares the approaches taken by three (COCOMO II, FP, UCP) widely used models for software cost and schedule estimation to develop COTECOMO (CONstructive Test Effort COst Model). It also documents what we call the large variance problem that is a leading cause of cost model brittleness or instability.

In this paper we present a novel method for estimating OO software projects size exploiting UML diagrams and three steps COTECOMO procedure. The main features of our approach are:

1. the method can be totally automated;
2. it exploits all the principal UML diagrams in its estimation process;
3. its output is in terms of SLOC, UCP so can be compared with well known standard like FP.

In this very moment, we are setting an experiment to fix the formulas' constants we introduced in the method.

References

1. Lazić Lj, Velašević D, Mastorakis N (2003) A framework of integrated and optimized software testing process. In: WSEAS Conference, August 11-13, Crete, Greece, also in WSEAS TRANSACTIONS on COMPUTERS 2(1)
2. Lazić Lj, Velašević D (2004) Applying simulation and design of experiments to the embedded software testing process. *Software Testing, Verification and Reliability*, John Willey & Sons Ltd., 14(4):257-282
3. Lazić Lj, Mastorakis N (2005) RBOSTP: Risk-based optimization of software testing process-Part 2. *WSEAS Transactions on Information Science and Applications*, ISSN 1790-0832, 2(7):902-916
4. Lazić Lj, Mastorakis N (2007) A framework of software testing metrics – Part 1 and 2. In: WSEAS Engineering Education 2007 Multiconference, Agios Nikolaos, Crete Island, Greece, pp 23-28
5. Jones C (1998) *Estimating software costs*. McGraw-Hill
6. Nageswaran Suresh (2001) Test effort estimation using use case points. In: *Quality Week 2001*, San Francisco, California, USA, <www.Cognizant.com/>

- cogcommunity/presentations/Test_Effort_Estimation.pdf>
7. Carbone M, Santucci G (2002) Fast&&Serious: a UML based metric for effort estimation. In: Proceedings of the 6th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE'02), Spain
 8. Garmus D, Herron D (2001) Function point analysis. Addison-Wesley, ISBN 0-201-69944-3
 9. Veenendaal EPWM van, Dekkers T (1999) Test point analysis: a method for test estimation. In: Kusters R, Cowderoy A, Heemstra F, Veenendaal E van (eds) Project Control for Software Quality, Shaker Publishing BV, Maastricht, The Netherlands
 10. Boehm B (1981) Software engineering economics. Prentice Hall
 11. Boehm B (2000) Safe and simple software cost analysis. IEEE Software, September/October 2000, pp 14–17, Available from <http://www.computer.org/certification/beta/BoehmSafe.pdf>
 12. Boehm B, Horowitz E, Madachy R, Reifer D, Clark BK, Steece B, Brown AW, Chulani S, and Abts C (2000) Software Cost Estimation with Cocomo II. Prentice Hall
 13. Hall M, Holmes G (2003) Benchmarking attribute selection techniques for discrete class data mining. IEEE Transactions on Knowledge and Data Engineering 15(6):1437–1447
 14. Jorgensen M (2004) A review of studies on expert estimation of software development effort. Journal of Systems and Software 70(1-2):37–60
 15. Karner Gustav (1993) Resource estimation for objectory projects. Objective Systems SF AB
 16. Nagappan N, Williams L, Vouk M, Osborne J (2005) Early estimation of software quality using in-process testing metrics: a controlled case study. In: Third Software Quality Workshop, co-located with the International Conference on Software Engineering (ICSE 2005), pp 46-52
 17. Uemura T, Kusumoto S, Inoue K (1999) Function point measurement tool for UML design specification. In: Proceedings of the 6th Int'l IEEE Software Metrics Symposium, IEEE-CS Press, pp 62-69
 18. Anda Bente (2003) Improving estimation practices by applying use case models. <www.cognizant.com/cogcommunity/presentations/Test_Effort_Estimation.pdf>
 19. Anda Bente et al. (2005) Effort estimation of use cases for incremental large-scale software development. In: 27th International Conference on Software Engineering, St Louis, MO, pp 303-311
 20. Carroll Edward R (2005) Estimating software based on use case points. In: 2005 Object-Oriented, Programming, Systems, Languages, and Applications (OOPSLA) Conference, San Diego, CA
 21. Menzies Tim, Chen Zhihao, Hihn Jairus, Lum Karen (2006) Selecting best practices for effort estimation. IEEE Trans. on Soft. Eng. 32(11)
 22. Clemmons Roy K (2006) Project estimation with use case points. CrossTalk Feb.
 23. Carper Jones (2004) Software project management practices: failure versus

success. CrossTalk Oct.

Chapter 7

A semantics of core computational model for ODP specifications

Youssef Balouki, Mohamed Bouhdadi

Mohammed V University, Department of Mathematics & Computer Science, B.P. 1014 Av. Ibn Bettouta, Rabat Morocco

Abstract. The Reference Model for Open Distributed Processing (RM-ODP) defines a framework within which support of distribution, interoperability and portability can be integrated. However other ODP standards have to be defined. We treat in this paper the need of formal notation for concepts in the computational language. Indeed, the ODP viewpoint languages are abstract in the sense that they define what concepts should be supported not how these concepts should be represented. Using the denotational semantics in the context of UML we define in this paper syntax and semantics for core ODP structural concepts defined in the RM-ODP foundations part and in the computational language namely object action, computational object, and computational interface. These specification concepts are suitable for describing and constraining ODP computational viewpoint specifications.

Keywords. RM-ODP, Computational Objects, Computational Interfaces, Denotational Semantics, UML/OCL

7.1 Introduction

The rapid growth of distributed processing has led to a need for coordinating framework for the standardization of Open Distributed Processing (ODP). The Reference Model for Open Distributed Processing (RM-ODP)

[1-4] provides a framework within which support of distribution, networking and portability can be integrated. It consists of four parts. The foundations part [2] contains the definition of the concepts and analytical framework for normalized description of (arbitrary) distributed processing systems. These concepts are grouped in several categories. The architecture part [3] defines a framework comprising five viewpoints, five viewpoint languages, ODP functions and ODP transparencies

However, RM-ODP [3] only provides a framework for the definition of new ODP standards. These standards include standards for ODP functions [6-7]; standards for modelling and specifying ODP systems; standards for testing ODP systems. Also RM-ODP recommends defining ODP types for ODP systems [8].

In this paper we treat the need of formal notation of ODP viewpoint languages. We use the UML/OCL denotational meta-modelling, semantics to define semantics for structural specification concepts in ODP computational language. The part of RM-ODP considered consists of modelling and specifying concepts defined in the RM-ODP foundations part and concepts in the computational language.

The paper is organized as follows. Section 7.2 describes related works. Section 7.3 we introduce the subset of concepts considered in this paper namely the object model and main structural concepts in the enterprise language. Section 7.4 defines the meta-model for the language of the model: object template, interface template, action template, type, and role. The meta-model syntax for considered concepts consists of class diagrams and OCL constraints. Section 7.5 describes the UML/OCL meta-model for instances of models Section 7.6 makes the connection between models and their instances using OCL. A conclusion end perspectives end the paper.

7.2 Related work

The languages Z [9], SDL [10], LOTOS [11] and, Esterel [12] are used in RM-ODP architectural semantics part [4] for the specification of ODP concepts. However, no formal method is likely to be suitable for specifying every aspect of an ODP system. In fact, these methods have been developed for hardware design and protocol engineering. The inherent characteristics of ODP systems imply the need to integrate different specification languages and different verification methods.

Elsewhere, this challenge in formal methods world is the same in software methods. Indeed, there had been an amount of research for applying the Unified Modelling Languages UML [13] as a notation for the defini-

tion of the syntax of UML itself [14-16]. This is defined in terms of three views: the abstract syntax, well-formedness rules, and modeling elements semantics. The abstract syntax is expressed using a subset of UML static modelling notations that is class diagrams. The well-formedness rules are expressed in Object Constrains Language OCL [17]. OCL is used for expressing constraints on object structure which cannot be expressed by class diagrams alone. A part of UML meta-model itself has a precise semantics [18, 19] defined using denotational meta-modelling semantics approach. A denotational approach [20] is realized by a definition of the form of an instance of every language element and a set of rules which determine which instances are and are not denoted by a particular language element. The three main steps of the approach are: (1) define the meta-model for the language for models (syntax domain), (2) define the meta-model for the language of instances (semantics domain), and (3) define the mapping between these two languages

Furthermore, for testing ODP systems [2-3], the current testing techniques [21, 22] are not widely accepted and especially for the enterprise viewpoint specifications. A new approach for testing, namely agile programming [23, 24] or test first approach [25] is being increasingly adopted. The principle is the integration of the system model and the testing model using UML meta-modelling approach [26, 27]. This approach is based on the executable UML [28]. In this context OCL is used to specify the invariants [19] and the properties to be tested [24]. The OCL invariants are defined using the UML denotational semantics and OCL itself.

In this context we used the meta-modelling syntax and semantics approach in the context of ODP languages. We defined syntax of a sub-language for the ODP QoS-aware enterprise viewpoint specifications [29]. We also defined a UML meta-model semantics for structural concepts in ODP enterprise [30].

7.3 The RM-ODP

We overview in this section the core structural concepts for ODP enterprise language. These concepts are sufficient to demonstrate the general principle of denotational semantics in the context of ODP viewpoint languages.

7.3.1 The RM-ODP foundations part

The RM-ODP object model [3] corresponds closely to the use of the term data-model in the relational data model. To avoid misunderstandings, the RM-ODP defines each of the concepts commonly encountered in object oriented models. It underlines a basic object model which is unified in the sense that it has successfully to serve each of the five ODP viewpoints. It defines the basic concepts concerned with existence and activity: the expression of what exists, where it is and what it does. The core concepts defined in the object model are object and action.

An object is the unit of encapsulation: a model of an entity. It is characterized by its behavior and, dually, by its states.

Encapsulation means that changes in an object state can occur only as a result of internal actions or interactions.

An action is a concept for modeling something which happens. ODP actions may have duration and may overlap in time. All actions are associated with at least one object: internal actions are associated with a single object; interactions are actions associated with several objects.

Objects have an identity, which means that each object is distinct from any other object. Object identity implies that there exists a reliable way to refer to objects in a model. When the emphasis is placed on behavior an object is informally said to perform functions and offer services, these functions are specified in terms of interfaces. It interacts with its environment at its interaction points which are its interfaces. An interface is a subset of the interactions in which an object can participate. An ODP object can have multiple interfaces. Like objects, interfaces can be instantiated.

The other concepts defined in the object model are derived from the concepts of object and action; those are class, template, type, subtype/supertype, subclass/superclass, composition, and behavioral compatibility.

Composition of objects is a combination of two or more objects yielding a new object.

An object is behaviorally compatible with a second object with respect to a set of criteria if the first object can replace the second object without the environment being able to notice the difference in the object behavior on the basis of the set of criteria.

A type (of an $\langle x \rangle$) is a predicate characterizing a collection of $\langle x \rangle$ s. Objects and interfaces can be typed with any predicate. The ODP notion of type is much more general than of most object models. Also ODP allows ODP to have several types, and to dynamically change types.

A class (of an $\langle x \rangle$) defines the set of all $\langle x \rangle$ s satisfying a type. An object class, in the ODP meaning, represents the collection of objects that sa-

tisfy a given type. ODP makes the distinction template and class explicit. The class concept corresponds to the OMG extension concept; the extension of a type is the set of values that satisfy the type at a particular time. A subclass is a subset of a class. A subtype is therefore a predicate that defines a subclass. ODP subtype and subclass hierarchies are thus completely isomorphic.

A $\langle x \rangle$ template is the specification of the common features of a collection x in a sufficient detail that an x can be instantiated using it.

Types, classes, templates are needed for object, interface, and action.

7.3.2 The RM-ODP computational language

The definition of a language for each viewpoint describes the concepts and rules for specifying ODP systems from the corresponding viewpoint. The object concepts defined in each viewpoint language are specializations of those defined in the foundations part of RM-ODP.

A computational specification defines the functional decomposition of an ODP system into objects which interact at interfaces. The basic concepts of the computational language are the computational interface, the computational object, the interaction and the binding object. The binding object is a computational object which supports a binding between a set of other computational objects.

An interaction is a signal, an operation or a flow.

A signal is an atomic shared action resulting in one-way communication from an initiating object to a responding object.

An operation is an interaction between a client object and a server object.

A flow is an abstraction of a sequence of interactions resulting in a conveyance of information from a producer object to a consumer object.

Like interaction kinds, an interface is signal, operation or flow.

A signal interface is an interface in which all the interactions are signals.

In an operation interface all the interactions are operations.

All the interactions are flows in a flow interface.

A computational object template comprises a set of computational interface templates which the object can instantiate, a behavior specification and an environment contract specification. The behavior of the object and the environment contract are specified in terms of a set of properties (attributes).

A computational interface template is associated to each kind of interface. It comprises a signal, a stream, or an operation interface signature as appropriate, a behavior specification and an environment contract specifi-

cation. Like for the object template, the behavior and the environment contract are defined as a set of properties.

7.4 The syntax domain

We define in this section the meta-models for the concepts presented in the previous section. Figure 7.1 defines the context free syntax for the core object concepts and Figure 7.2 defines the context free syntax for concepts in the computational language.

There are several context constraints between the syntax constructs. In the following we define some of these context constraints using OCL for the defined syntax.

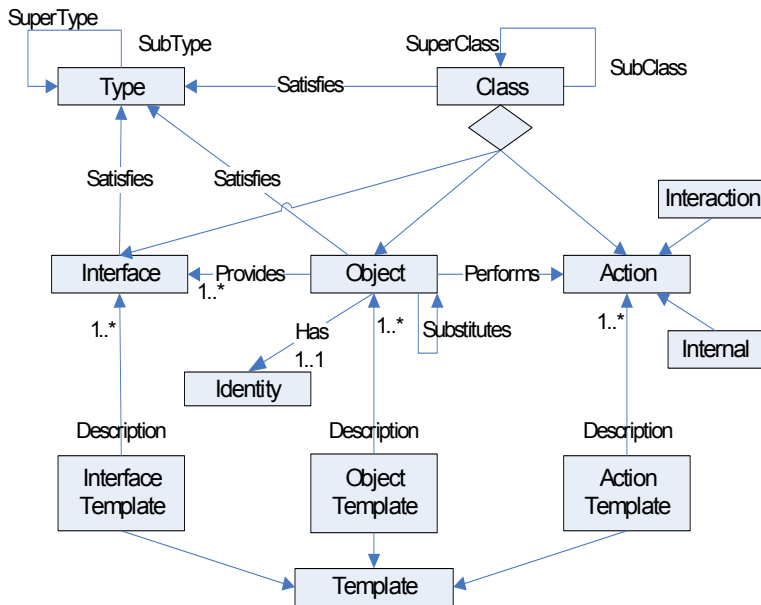


Fig. 7.1. RM-ODP Foundation Object Model

In the following we define context constraints for the defined syntax.

Context m: Model **inv:**

```
m.Roles->includesAll(m.Roles.Source->union(m.Roles.Target))
m.Roles->includesAll(m.ObjectTemplates.Roles)
m.Roles->includesAll(m.Interactiontemplate.roles)
m.Roles->includesAll(m.InterfaceTemplate.roles)
m.InteractionTemplates->includesAll(m.ObjectTemplates.Interactiontemplates)
m.InteractionTemplates.Types->includesAll(m.Types)
```

```

m.ObjectTemplates.InterfaceTemplates->includesAll(m.InterfaceTemplates)
m.ObjectTemplates.InterfaceTemplates->includesAll(m.InterfaceTemplates)
includesAll(m.ObjectTemplates.Interactiontemplates)

```

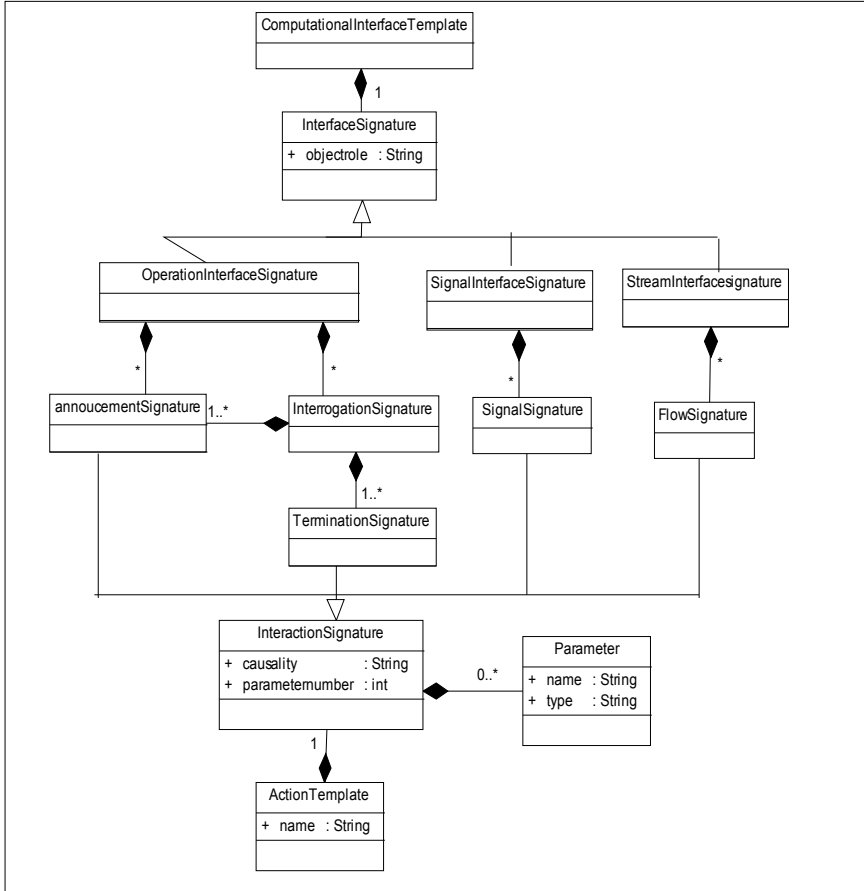


Fig. 7.2. Concepts in RM-ODP Computational Language

7.5 Semantics domain

The semantics of a UML model is given by constraining the relationship between a model and possible instances of that model. That is, constraining the relationship between expressions of the UML abstract syntax for models and expressions of the UML abstract syntax for instances. The latter constitute the semantics domain and is integrated with the syntax do-

main within the same meta-model (Figures 7.1 and 7.2). This defines the UML context free abstract syntax. We give in the following a context constraint between the instances of semantics domain. These constraints are relatively simple but demonstrate the general principle.

We consider the concepts of subtype/supertype (RM-ODP 2-9.9) and subclass/superclass (RM-ODP 2-9.10) as relations between types and classes correspondingly.

Context *m* : model **inv**

m.types-> forall(*t1*: Type, *t2*: Type | *t2*.subtype -> includes(*t1*) implies *t1*.valid_for.satisfies_type=*t2*)

m.types-> forall(*t1*: Type, *t2*: Type | *t1*.supertype -> includes(*t2*) implies *t1*.valid_for.satisfies_type=*t2*)

7.6 Meaning function

The semantics for the UML-based language defined focuses on the relationship between a system model and its possible instances (systems). Both of the domains are defined using UML/OCL. The association between the instances of the two domains (syntax and semantics) is defined using the same UML meta-models. The OCL constraints complete the meaning function the context of UML [20, 31]. We give in the following some constraints which are relatively simple, but they demonstrate the general principle. We use the semantics of invariants as defined in [19].

Clause 7.1.9 of part3 “computational object template: An object template which comprises a set of Computational interface template which the object can instantiate, a behaviour specification and an environment contract specification.”

Context *m* : Model **inv** :

m.ComputationalObjectTemplate->

includesall(*m*.objectTemplate.Computational.InterfaceTemplate)->

union (*m*.behaviorTemplate)

Clause 7.1.10 of part3 “computational interface template: An interface template for either a signal interface, a stream interface or operation interface. A computational interface template comprises a signal, a stream or an operation signature as appropriate, a behaviour specification and an environment contract specification.”

Context *m* : Model **inv** :

m.ComputationalInterfaceTemplate

->includesall(*m*.objectTemplate.Signal.InterfaceTemplate)

```

-> union (m.objectTemplate.StreamInterfaceTemplate)
-> union (m.objectTemplate.OperationInterfaceTemplate)
-> union (m.behaviorTemplate)

```

Each interface has a signature:

Context cit: ComputationelInterfaceTemplate **inv** :
cit.OperatioInterfaceTemplate -> inclaudeAll(cit.AnnoucementSignature)
->union(cit.IntegrationSignature)
and cit.SignalInterfaceTemplate -> inclaudeAll(cit.SignalSignature)
and cit.StreamInterfaceTemplate -> inclaudeAll(cit.StreamSignature)

Clause 7.1.11 of part3 “signal interface signature: An interface signature for signal interface . A signal interface signature comprises a finite set of action template, one for each signal type in interface.

Each action template comprises the name for signal, the number, names and types of its parameters and indication of causality (initiating or responding, but not both) with respect to the objects which instantiates the template.

Context: Signal InterfaceTemplate **inv** :
self.signaltype -> forall(t | self.siganture->includes(set (t.actiontemplate))) and
Forall(at : actiontemplate | at.names ->includesall(self.signal.name)-
>union(self.parmeter.name) and
at.types ->inclaudeall(self.parmeter.type)) and
(self.causality ==#Intiating xor self.causality ==#Responding)

Clause 7.1.12 of part3 “operation interface signature: An interface signature for operation interface. A operation interface signature Comprises a set of announcement and interrogation signatures as appropriate, one for each operation type in the interface, together with an indication of causality (client or server, but not both) for the interface as whole, with respect to the object which instantiates the template. Each announcement signature is an action template containing the name of the invocation and the number, names and types of its parameters.

Context OperationInterfaceTemplate **inv** :
Sel.ocllsTypeOf (Signal InterfaceTemplate)
self.signature -> includes (set(announcementsignature)) -> union
set (integrationsignature)
self.announcementsignature -> includes(set (actiontemplate)) and
Forall(at : actiontemplate | at.names ->includesall(self.invocation.name)
->union(self.parmeter.name) and
at.types ->inclaudeall(self.parmeter.type))

Each interrogation signature comprises an action template with the following elements: 1) the name of the invocation; 2) the number, names and

types of its parameters; and 3) a finite, non-empty set of action templates, one for each possible termination type of invocation each containing both name of the termination and the number, names and types of its parameters.

Context InterrogationInterface **inv**:

```
self.signature -> includes (set(actiontemplate)) and Forall( at : actiontemplate |
at.names ->includesall(invocation.name) ->union(parameter.name) and at.types -
->incluadesall(parameter.type)) and forall (I:invocation | exists( at :actiontemplate
|at.names ->includesall(I.termination.name)->union(termination.parameter.name)
and at.types ->incluadesall(termination.parameter.type))
```

Context OperationTntemplate **inv** :

```
Self.interogationTemplate -> includes(set(self.InvocationTemplate)->union
(self.TerminationTemplate) and Forall( I:invocationTemplate |exists(T : termina
tiontemplate | self.I.possible_termination = T)
```

Clause 7.1.13 of part 3 “stream interface signature: An interface signature for stream interface. A stream interface comprises a set of action templates, one for each flow type in the stream interface. Each action template for a flow contains the name for flow, the information type of the flow, and indication of causality for flow (i.e. producer or consumer but not both) with respect to the object which instantiates the template.

Context Stream InterfaceTemplate **inv** :

```
Self.ocllsTypeOf (Signal InterfaceTemplate )
Self.streamtype -> forall(t | includes(Self.signature -> set (t.actiontemplate)) ) and
Forall( at : actiontemplate | at.names ->includesall(self.flow.name) and
at.types ->incluadesall(self.inforamtion.type)) and (self.causality ==#Producer xor
self.causality ==#Consumer)
```

7.7 Conclusion

We address in this paper the need of formal ODP viewpoint languages. Indeed, the ODP viewpoint languages define what concepts should be supported not how these concepts should be represented. One approach to defining the formal semantics of a language is denotational: elaborating the value or instance denoted by an expression of the language in a particular context. Using the denotational meta-modeling semantics, we define in this paper the UML/OCL based syntax and semantics for a core Open Distributed Processing concepts defined in the foundations part and in the computational viewpoint language. These concepts are suitable for describing and constraining ODP computational viewpoint specifications. We are applying the same denotational semantics to define semantics for concepts

characterizing dynamic behavior in the enterprise and computational languages.

References

1. ISO/IEC (1994) Basic reference model of open distributed processing - Part 1: Overview and guide to use. ISO/IEC CD 10746-1
2. ISO/IEC (1994) RM-ODP-Part 2: Descriptive model. ISO/IEC DIS 10746-2
3. ISO/IEC (1994) RM-ODP-Part 3: Prescriptive model. ISO/IEC DIS 10746-3
4. ISO/IEC (1994) RM-ODP-Part 4: Architectural semantics. ISO/IEC DIS 10746-4
5. OMG (1991) The object management architecture. OMG
6. ISO/IEC (1999) ODP type repository function. ISO/IEC JTC1/SC7 N2057
7. ISO/IEC (1995) The ODP trading function. ISO/IEC JTC1/SC21
8. Bouhdadi M et al. (2000) An informational object model for ODP applications. *Malaysian Journal of Computer Science* 13(2):21-32
9. Spivey J.M (1992) The Z reference manual. Prentice Hall
10. IUT (1992) SDL: Specification and Description Language. IUT-T-Rec. Z.100
11. ISO/IUT (1998) LOTOS: A formal description technique based on the temporal ordering of observational behavior. ISO/IEC 8807
12. Bowman H et al (1995) FDTs for ODP. *Computer Standards & Interfaces Journal*, Elsevier 17(5-6):457-479
13. Rumbaugh J et al (1999). The unified modeling language. Addison Wesley
14. Rumpe B (1998) A note on semantics with an emphasis on UML. In: Second ECOOP Workshop on Precise Behavioral Semantics, Springer LNCS 1543:167-188.
15. Evans A et al. (1998) Making UML precise. *Object Oriented Programming, Systems languages and Applications*, (OOPSLA'98), Vancouver, Canada, ACM Press
16. Evans A et al. (1999) The UML as a Formal Modeling Notation. LNCS 1618, Springer, pp 349-364
17. Warmer J, Kleppe A (1998) The object constraint language: Precise modeling with UML. Addison Wesley
18. Kent S et al. (1999) A meta-model semantics for structural constraints in UML. In: Kilov H, Rumpe B, Simmonds I (eds) *Behavioral specifications for businesses and systems*, Kluwer, chapter 9
19. Evans E et al. (1999) Meta-modeling semantics of UML. In: Kilov H, Rumpe B, Simmonds I (eds) *Behavioral specifications for businesses and systems*, Kluwer, chapter 4.
20. Schmidt DA (1986) Denotational semantics: A methodology for language development. Allyn and Bacon, Massachusetts
21. Myers G (1979) The art of software testing. John Wiley & Sons
22. Binder R (1999) Testing Object Oriented Systems. Models. Patterns and Tools. Addison-Wesley

23. Cockburn A (2002) Agile software development. Addison-Wesley
24. Rumpe B (2004) Agile Modeling with UML. Springer LNCS 2941:297-309
25. Beck K (2001) Column on test-first approach. IEEE Software 18(5):87-89
26. Briand L (2001) A UML-based Approach to System testing. Springer LNCS 2185:194-208
27. Rumpe B (2003) Model-Based Testing of Object-Oriented Systems. Springer LNCS 2852:380-402
28. Rumpe B (2002) Executable modeling UML. A vision or a nightmare? In: Issues and Trends of Information technology management in Contemporary Associations, Seattle, Idea Group, London, pp 697-701
29. Bouhdadi M et al. (2002) An UML-based meta-language for the QoS-aware enterprise specification of open distributed systems. Collaborative Business Ecosystems and Virtual Enterprises, FIP Series, Springer 85:255-264
30. Bouhdadi M, Balouki Y, Chabbar E (2007) Meta-modelling syntax and semantics for structural concepts for open networked enterprises. LNCS 4707:45-54.
31. France R, Kent S, Evans A, France R (1999) What does the term semantics mean in the context of UML. In: Moreira Ana MD, Demeyer S (eds) ECOOP Workshops 1999, ECOOP'99 Workshop Reader, Springer, LNCS 1743:34-36

Chapter 8

Computer simulation via direct modeling

Alex Soiguine

SOiGUINE LLC, 1333 N. Camino Alto, #227, Vallejo, CA 94589, USA

Abstract. The article introduces a design of a novel computer simulation environment that is based on an original approach, a method of direct computer modeling. This “triple helix”-like method is comprised of basic physical laws of element interactions, data type class hierarchy of an object-oriented computer language, and an executing computer program that can potentially run on a network of parallel processors. The approach is universally applicable for modeling a wide spectrum of physical, and other, processes and phenomena. The method is directed towards consistent elimination of solving equations as a main object of computer numerical algorithms in the simulation problem. New computer simulation for complex adaptive systems of interactive parts that change in response to local inputs will be used in tight conjunction with the C++ object-oriented computer language.

Keywords. Computer simulation, Parallel algorithms, Object-oriented languages, Numerical methods

8.1 Introduction

Computer modeling and simulation are the key elements in achieving progress in engineering and science. New simulation components require complex algorithms and must function efficiently on an evolving range of architectures designed for large-scale parallel computations.

The current work is about bringing something positive and practical into that troubled area. The proposed method is directed at consistent elimination of simulation methods that have evolved over the last 300 years from sequential pencil-and-paper procedures as a main object of computer numerical algorithms in physical processes simulation problems. The objective is to demonstrate the feasibility of the Direct Computer Modeling paradigm for developing modern computational, numerical techniques and sophisticated computer modeling methods with the simulation capability for a wide spectrum of physical and other processes. The Direct Computer Modeling (DiCOMO) paradigm has, in its very nature, potential intrinsic parallelism of the numerical algorithms, implying easy and direct application to high performance computing. The DiCOMO method is about new behavior-oriented computations supported by contemporary software and hardware resources. It is an elegantly simple mathematical approach to capturing physics, making computer algorithms adequate for evaluating real world phenomena. The DiCOMO approach to creating computer programs for simulating physical phenomena can be successfully applied to an entire spectrum of multi-physics systems. This work demonstrates the ability of the DiCOMO to explicitly define the necessary class type variables corresponding to truly involved physical entities and ways to establish their communication, taking protocols as class variables pointer structures, and embedding the overall physical processes into a complex adaptive system of interactive parts and elements, which behave and change their states in response to local inputs defined by their neighbors' interactions.

A. Soiguine formulated the original idea in "The Method of Direct Computer Modeling" [1]. It was successfully applied to some computer simulation problems in traditional single processor computational environment and conventional programming languages [2]. The DiCOMO approach to simulation of physical processes is an effective alternative to numerical methods based on centuries-old ideas. A physical process model in the DiCOMO context is "triple helix"-like, comprising of basic physical laws of element interactions, data type class hierarchy of an object-oriented computer language, and hardware, executing computer program that can run on a network of parallel processors.

8.2 Background

There is an acute need for developing an effective computer simulation, taking advantage of the parallel nature of the proposed numerical schemes

for easy and direct application to high performance computing. To realize that the DiCOMO can successfully and very effectively work in creating necessary software environment, we are considering its potential implementations in some practically important examples.

8.2.1 The DiCOMO general scheme as applied to continuous media

PDEs remain the basic models of real physical phenomena in continuous media, while computers play the role of powerful resources that numerically solve equations.

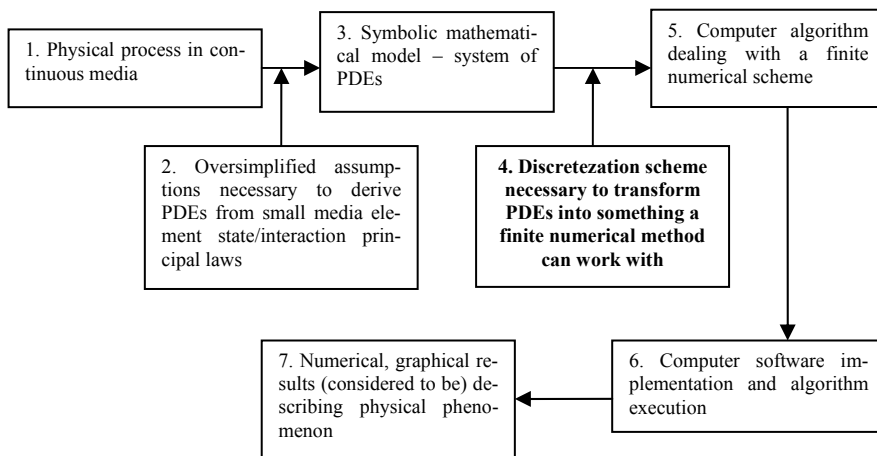


Fig. 8.1. The logic of the traditional approach

Figure 8.1 shows the logic of the traditional approach. PDEs are derived through a discretization procedure of real physical media, with a subsequent mathematical limit process, accompanied generally by many non-physical assumptions. Any PDE is just an expression that symbolically encodes an unnecessarily oversimplified process that is meant to portray a real physical process. Would it not be more logical to run a discrete algorithm of interactions/state evolutions of small media elements? In other words, it is more physically consistent and computationally effective to exclude blocks 3 and 4 from the traditional scheme and remove the statement “oversimplified assumptions necessary to derive PDE from” from block 2.

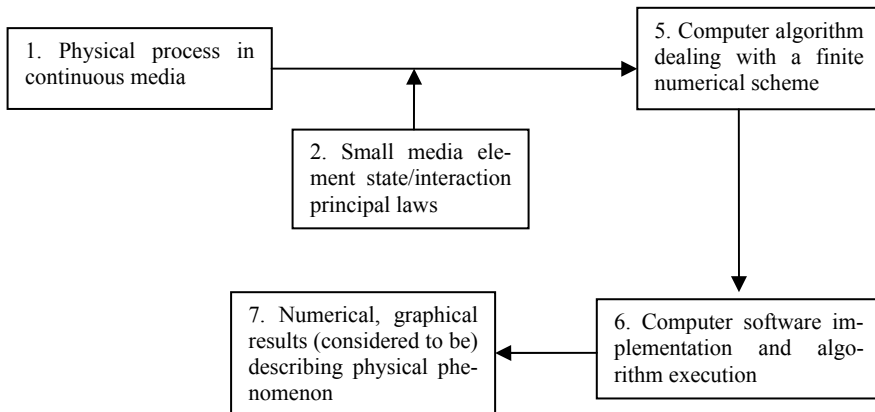


Fig. 8.2. The logic of the modified traditional approach

An option exists to use a large array of microprocessors, corresponding to elements in a physical medium [1]. Each microprocessor would save all numerical parameters defining the state of an element, all functions necessary to transform the element state parameters, and sufficient number of ports to communicate with processors within the network. In that case, the array of processors should behave exactly like a real physical medium. To some extent, it would resemble an analog computer but with much greater capabilities. At the same time, all this can be formulated on a software level using object-oriented class type variables hierarchy. For technical details, including resolving the algorithms stability problem, see [3-6].

Elimination of PDEs from the continuous media numerical algorithms and replacing the equation discretezation scheme with the original discretezation of the medium allows to eliminate at least two computational disadvantages of the traditional approach. First, when solving a PDE, distribution of equations solution over many processors is not straightforward. The sparse matrix fill-in issue may cause irresolvable memory requirements. Confusing terminology like “finite element analysis” has nothing to do with small medium interacting elements. “Finite elements” means special classes of simple functions used to approximate unknown solution of a PDE. In that method, data inevitably must be shared among processors, causing the second problem: data traffic, resulting from data sharing, fully diminishes positive effects of parallelization.

One other approach should be mentioned in connection with the current proposal: the cellular automata approach [7]. Although some expectations have existed about the applicability of that method to continuous media simulation with multiprocessor parallelization [1], it did not realize to be a proper tool. DiCOMO is much more general and promising because it does

not suffer from the cellular automata principal assumption: a “cell” should be a device with a finite number of states.

8.2.2IC simulation algorithms

Traditional substitution of real flows of physical processes with hieroglyphic scripts and using the latter as objects of computer simulation, instead of simulation of a phenomenon as is, follows from historical reasons and often brings tough numerical problems that have nothing to do with physical reality. A typical example of that is transformation of Kirhhoff’s laws of electric circuits into systems of algebraic equations. Technical mathematical problems appearing in sparse matrix solution algorithms do not reflect real electrical circuit behavior. The C++ code shown below demonstrates how conservation of sum of currents at nodes and electromagnetic energy conservation, along with parameter transformation law, can be represented in the DiCOMO simulation scheme. There is no algebraic system at all, to say nothing of terrific numerical simplicity. Circuit elements exchange information about their states through ports. The states are iteratively adjusted to get final matching to the states at some predefined (fixed) elements (two voltage nodes in the current example).

The base class for an arbitrary electronic device is formally represented by a C++ class type variable:

```
class el_dev
{
public:
    string el_deva_type;      // like "RES" for resistors, "CAP"
                             // for capacitors, "TRA" for
                             // transistors, etc. (not as
                             // critical as for reading SPICE
                             // deck)
    string el_deva_name;     // mainly, el_deva_type plus index
                             // (number)
    long instN;              // device number (index)
    long nmbPorts;           // number of ports
    state* el_deva_par;      // name and value for such items as
                             // resistor resistance, capacitor
                             // capacitances, temperatures, etc.
    state* el_deva_stapar;    // name and value for physical
                             // entities defining current device
                             // state, say, drain current,
                             // voltage at gate (to some extend,
                             // duplicated by port_par values)
    port* el_deva_port;      // array of the device ports
    port* empty_port;

    el_dev(void);
```

```
~el_dev(void);

bool equal(port p1, port p2);
    // checks if ports are equal (just
    // by name and owner's name
port* get_port(char* name);
    // returns pointer to a port by its
    // name
int connect2(el_dev* neighbor, string mine, string his);
    // connects current port (name "mine")
    // to a "neighbor"'s device port of name "his"
double get_port_curr(string mine);
    // returns electric current at the
    // current device's port
    // named "mine"
double get_port_volt(string mine);
    // returns voltage at that port
double get_2port_curr(string mine);
    // returns the sum of all electric
    // currents at port "mine"
    // from all ports it is connected to

virtual state* get_new_state() = 0;
};
```

The *state* and *port* structures are defined as:

```
typedef struct astate    // structure to keep all information
                        // about numerical parameter
{
    string sta_name;      // name of parameter
    double sta_value;     // its numerical value
    bool set;             // has been set
    bool fixed;           // must be kept as given
} state;

typedef struct aport
{
    string port_name;     // may be like "RIGHT" or "LEFT" for
                        // resistor or capacitor, or "SOURCE",
                        // "GATE", "DRAIN"
                        // for transistor, etc.

    string el_deva_own;   // name of the device port belongs to,
                        // that's "el_deva_name" comprising
                        // device type and index (number)

    state* port_par;      // what physical entities may be read
                        // and saved at the port: mainly, at the
                        // port and exiting (entering) current

    aport* port_nei;      // array of ports the current port talks to
    long nmbNei;          // number of other ports it is connected to
```

```
bool connected; // is connected?
} port;
```

Defined class type variables can fully describe arbitrary network device states, and interconnect topologies and communication protocols.

The class, representing a resistor, inherits `el_dev` and acquires extra functions. With the necessary implementations of the member functions the whole simulation program for the resistor chain connecting two given voltage sources works through the code containing just several lines and only executing simple arithmetical operations:

```
for (int i=0;i<NUM_RES;i++)
{
    if (resres[i]->get_port("LEFT")->port_par[0].fixed)
        l=resres[i]->get_port_volt("LEFT");
    if (resres[i]->get_port("RIGHT")->port_par[0].fixed)
        r=resres[i]->get_port_volt("RIGHT");
}

double act;
act = r-l;

do
{
    resid = 0.0;
    double re = 0.0;
    for (int i=0;i<NUM_RES;i++)
    {
        re = resres[i]->get_new_state();
        if (re!=BAD_DOUBLE)
            resid+=re;
    }

    resid -=l;
    resid = act/resid;
    for (int i=0;i<NUM_RES;i++)
    {
        resres[i]->scale_curr_val(resid);
    }
} while (fabs(1-resid)>1.0e-20);
```

Only two passes between given (defined) voltage nodes are necessary. Although the example is quite simple, it contains all features that should be used for an arbitrary device network. The important ideas of the DiCOMO implementation in the case of electrical circuits, and any boundary problem, are to define paths between predefined elements, propagate states between predefined sets of elements by state transformation laws and adjust elements state parameters at defined nodes (boolean parameter *fixed* in the *state* structure). In the case of ICs, the Kirhhoff's laws support stability. In the case of continuous media we need to use time step adjustments by applying energy conservation conditions. We see in this example that a sys-

tem (network) evolves by exchanging information about elements states through their ports, rapidly achieving a stable overall network state.

All that remains to do in the case of a network of arbitrary electronic devices is specific implementation of functions `get_new_state()` for each type of a device. When an integration scheme is necessary inside that function, for example in the cases of active devices or transistors or transistors containing internal devices like nanoobjects, a specific relation between overall and device specific click timers will be used. That idea of factorizing a sequence of clicking timers is also very specific and powerful in the DiCOMO simulation scheme.

Comparing the speed of the above code with publicly available SPICE3 code in the case of 100 serial resistors in the loop of 1000 cycles showed that DiCOMO runs about 30 times faster.

The extension of the C++ language followed from the DiCOMO, when applied to integrated circuit simulation, does not have the restrictions of FPGA (or FPOA) languages, which resulted from their applicability to only specific and simple device types. On the opposite side of the spectrum of networks of arbitrary devices, the C++ language extension is significantly more flexible than SystemC[®], which attempts to work in similar situations [8].

8.2.3 The molecular binding simulation

An intriguing application for the DiCOMO approach is the “molecular Lego” – simulation of combining molecules into complex structures. It is worth noticing that parallelization in DiCOMO is quite different from the parallel algorithms appearing in the PDE solutions [9].

The idea is to consider, with the help of DiCOMO approach, a simulation of capturing metal atoms by carbon nanotube structure. The resulting molecular composition may be used for hydrogen storage. The main requirement set by DOE is that the storage capacity must be above 6wt% of hydrogen. Preliminary first-principle calculations, based on quantum mechanical density functional theory [10], demonstrated that a high-transition metal, such as Ti, absorbed on a single wall carbon nanotube, can dissociate hydrogen molecules and strongly bond four H₂ molecules, creating a reversible system with up to 8wt% of hydrogen.

The principal difference between the DiCOMO approach for that kind of processes and previously considered examples is that connections between interacting elements change with time. A molecule, as a 3D element with directed vectors of interaction forces, will have in the C++ class type representation the `aport* port_nei` array from the structure `port`, va-

rying in time, depending of the molecular orientation in space. Technically, it will be a more difficult programming task, although nothing unexpected for the approach applicability.

8.2.4 DiCOMO approach compatibility with the Sun's Fortress object-oriented language for parallelization

As stated in "The Method of Direct Computer Modeling" [1], the ideal situation for the DiCOMO implementation would be a multiprocessor system with the number of CPUs equal to the number of interacting elements, representing the modeled physical media. Since multiprocessor systems are becoming more and more available, an urgent demand for an adequate programming language arises.

Sun Microsystems's new language, Fortress, is a part of an effort to come up with a design for the supercomputer of the future through the program on High Productivity Computing Systems. The language has been under development for more than three years. The most important thing in Fortress is that it is parallel. Most popular languages, such as Fortran and C++, have no direct concurrency, which must be grafted on top of them. With Fortress, program structure will reflect the science. The language machine assumptions include features critically important to the DiCOMO paradigm: not worrying about the number of processors, vast amount of parallelism (millions of simultaneous threads), fine-grained threading (any loop is parallel by default), shared global address space, transactional access to shared variables. Fortress will be offered to the community at large via an open source format. DiCOMO will profit immensely from the existence of Fortress.

8.3 Conclusions

The DiCOMO object-oriented based approach allows the development of algorithms that, being parallel in nature, are independent of a platform multiprocessor/network designs and node-CPU architecture. The abstractions that represent physics of the simulation remain untouched. One can think of the approach as an inverse of C. Mead's "neuromorphic" network modeling approach [11]. Physical interactions between elements and element states evolution are reproduced on the level of object-oriented class type variable states/interconnections/communication and communication protocol implementations.

The overall program is to design a computer simulation environment that would be universally applicable for modeling a wide spectrum of physical, and other, processes and phenomena based on an original approach, a method of direct computer modeling, tightly connected to object-oriented computer languages, physically natural model representations and potentially straightforward implementations in parallel processor environments. The specific objectives of the program are as follows:

1. Create templates of basic C++ class type variables representing general physical media entities following the DiCOMO philosophy.
2. Create derived class hierarchy representing different types of physical systems: general continuous media, electrical circuits, transistors bearing nano-scale elements (carbon nanotubes) and molecular dynamics with special attention to molecular binding problems.
3. Demonstrate feasibility of the approach to simulation of networks comprising traditional devices (passive, MOSFETs, etc.).
4. Demonstrate superiority of the approach in speed/number of operations compared with traditional matrix solution based simulation tools, such as SPICE.
5. Demonstrate feasibility of the approach to simulation of networks of novel types of transistors comprising nanoscale elements.
6. Check feasibility of the method in applying to the “molecular Lego” problems.
7. Formulate basic problems of the standardization of C++ language extension necessary for effective applicability of the DiCOMO paradigm.

References

1. Alexandrov VV, Soiguine AM (1989) The method of direct computer modeling. Preprint #102, Institute of Informatics of the USSR Academy of Science (in Russian), <http://www.soiguine.com/dicom.pdf>
2. Backus J (1978) Can programming be liberated from the von neumann style? A functional style and its algebra of programs. *Comm. of the ACM* 21:613-641
3. Soiguine A (2004) The direct computer modeling approach to continuous media dynamics: C++ implementation. In: *Proceedings of the 1st International Conference “From Scientific Computing to Computational Engineering”*, Athens, <http://www.soiguine.com/DirectComputerModelingApproach.pdf>
4. Soiguine A (2005) The direct computer modeling paradigm applied to string simulation: An example of intellectual medium. In: *1st International Conference on Experiments/Process/System Modeling/Simulation/Optimization*.

-
- Athens, Greece, <http://www.soiguine.com/DirectComputerModelingAppliedToString.pdf>
5. Soiguine A (2005) Continuous media simulation in C++ class type variables environment. In: 3rd WSEAS International Conference on Heat Transfer, Thermal Engineering & Environment. Corfu, Greece, http://www.soiguine.com/Continuous_Media_Simulation.pdf
 6. Soiguine A (2007) Direct computer modeling – New paradigm for continuous media simulation. Accepted for publication in Journal of Fluid Dynamics (Riadh, KSA), <http://www.fluidynamics.edu.sa>
 7. Wolfram S (2002) A new kind of science. Wolfram Media Inc.
 8. IEEE Standard SystemC[®] Language Reference Manual (2006)
 9. Plimpton S, Hendrickson B (1995) Parallel molecular dynamics algorithms for simulation of molecular systems. In: T. G. Mattson (ed) Parallel Computing in Computational Chemistry, pp 114-136
 10. Le Bris C, Lions P-L (2005) From atoms to crystals: A mathematical journey. Bulletin of the American Mathematical Society 42:291-363
 11. Mead C (1989) Analog VLSI and neural systems. Addison-Wesley

Chapter 9

Image segmentation with improved artificial fish swarm algorithm

Mingyan Jiang,^{1,2} Nikos E. Mastorakis,³ Dongfeng Yuan,² Miguel Angel Lagunas¹

¹ Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), Av. Del Canal Olímpic s/n, 08860 Castelldefels (Barcelona), Spain
{mingyan.jiang, m.a.lagunas}@cttc.es

² School of Information Science and Engineering, Shandong University, Jinan, 250100 China, {jiangmingyan, dfyuan}@sdu.edu.cn

³ Military Institutes of University Education (ASEI), Hellenic Naval Academy, Terma Chatzikyriakou 18539, Piraeus, Greece,
mastor@softlab.ece.ntua.gr

Abstract. Some improved adaptive methods about step length are proposed in the Artificial Fish Swarm Algorithm (AFSA) which is a new heuristic intelligent optimization algorithm. The experimental results show that proposed methods have better performances such as good and fast global convergence, strong robustness, insensitive to initial values, simplicity of implementation, we apply the method in the image processing for the multi-threshold image segmentation compared with Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), the properties are discussed and analysed in the end.

Keywords. Adaptive step length, Multi-threshold, Image segmentation, AFSA, GA, PSO

9.1 Introduction

In recent years, with the rise of artificial intelligence and artificial life, the research of swarm intelligence aroused great concern of numerous scholars, and some new-type bionic algorithms with the swarm intelligence are become hot research topics such as the Genetic Algorithm (GA) [1], the Particle Swarm Optimization (PSO) algorithm [2], the Ant Colony Optimization (ACO) algorithm [3], the Bees Algorithm (BA) [4], and the Artificial Fish Swarm Algorithm (AFSA) [5]. Their applications in many kinds of scientific research fields show their good properties and practical value. They have some common characters and also have their unique characters. AFSA as a new optimization algorithm becomes a very hot topic, it offers new ideas to solve the optimization problem in signal processing [6], complex function optimization [7], neural network classifiers [8], network combinatorial optimization [9], multi-user detection in communication [10,11], sequence code estimation [12], and some applications [13]. In these applications, the algorithm reflects good performances and becomes a prospective method in solving optimization problems. Its basic idea is to imitate the fish behaviours such as praying, swarming, following with local search of fish individual for reaching the global optimum, it is random and parallel search algorithm, it has the good ability to overcome local extrema, obtain the global extrema and has fast convergence speed. In this paper, we propose some adaptive step methods and apply one in image processing.

9.2 The artificial fish swarm algorithm

9.2.1 The artificial fish

The Artificial Fish (AF) realizes external perception by its vision. X is the current state of a AF, $Visual$ is the visual distance, and X_v is the visual position at some moment. If the state at the visual position is better than the current state, it goes forward a step in this direction, and arrives the X_{next} state; otherwise, continues an inspecting tour in the vision. The greater number of inspecting tour the AF does, the more knowledge about overall states of the vision the AF obtains. Certainly, it does not need to travel throughout complex or infinite states, which is helpful to find the global optimum by allowing certain local optimum with some uncertainty.

9.2.2 Five Basic Behaviors of AF

Fish usually stay in the place with a lot of food, so we simulate the behaviors of fish based on this characteristic to find the global optimum, which is the basic idea of the AFSA. Five basic behaviors of AF are defined [5,12] as follows:

(1). AF_Prey: This is a basic biological behavior that tends to the food; generally the fish perceives the concentration of food in water to determine the movement by vision or sense and then chooses the tendency. (2). AF_Swarm: The fish will assemble in groups naturally in the moving process, which is a kind of living habits in order to guarantee the existence of the colony and avoid dangers. (3). AF_Follow: In the moving process of the fish swarm, when a single fish or several ones find food, the neighborhood partners will trail and reach the food quickly. (4). AF_Move: Fish swim randomly in water; in fact, they are seeking food or companions in larger ranges. (5). AF_Leap: Fish stop somewhere in water, every AF's result will gradually be the same, the difference of objective values (food concentration, FC) becomes smaller within some iterations, it might fall into local extrema, change the parameters randomly to the still states for leaping out current state. The detail behaviors pseudo-code can be seen in [6].

9.2.3 The improved adaptive step length in AFSA

In the AFSA, there are many parameters that have impacts on the final optimization result. In this paper, we only consider the parameter *Step*. With the increase of the *Step*, the speed of convergence is accelerated. However, when the increase of the *Step* is out of a range, the speed of convergence is decelerated, and sometimes the emergence of vibration can influence the speed of convergence greatly. Using the adaptive step may prevent the emergence of vibration, increase the convergence speed and enhance the optimization precision.

In the behaviors of AF_Prey, AF_Swarm and AF_Follow, which use the *Step* parameter in every iteration, the optimized variables (vector) have the various quantity of $Step \cdot \text{rand}()$, *Step* is a fixed parameter, $\text{rand}()$ is a uniformly distributed function. We give some adaptive *Step* methods as follows (*t* means iteration time).

1. $Step_{t+1} = \alpha \cdot Step_t$, $\alpha = 0.9 \sim 0.99$ and $Step_{t=1} = Step$
2. Prior period of iteration, using $Step \cdot \text{rand}()$; later period, using method (1).

$$3. \text{Step}_{t+1} = \frac{\beta \square N - t}{N} \text{Step}, \beta = (1.1 \sim 1.5), N \text{ is the all iteration times.}$$

The method (1) can insure the precise result when the iteration is in the optimal research range, but might trap in local extrema. Method (2) can balances above problems. Method (3) has a relation with the iteration time, and gradually decreases the *Step*, so decreases the various quantities of optimized variables in each iteration time. Using the adaptive step, we can select the *Step* more randomly which can guarantee the fast convergence, the result's precision and stability.

For example, using fixed *Step*=2 and adaptive step ($\beta=1.2$) of method (3), we optimize a nonlinear function with many local extreme points by AFSA and improved AFSA, the function $f(x, y) = x \sin(2\pi x) - y \sin(2\pi y) + 1$, and $x, y \in [-1, 2]$. The comparison result is shown in Figure 9.1. With the adaptive step, convergence speed is faster and the result is more stable especially when the optimized variables are many.

9.3 Image segmentation by improved AFSA

9.3.1 The multi-threshold image segmentation

The image segmentation is key pre-step for the object identification; there are many conventional segmentation methods such as the maximum variance between clusters method, maximum entropy (ME) threshold method, minimum error threshold method, etc. The principle of image segmentation based on ME [14] is: the image gray level $\in [0, l-1]$, t is the

threshold for the image segmentation, $P_t = \sum_{i=0}^t p_i$ and p_i is the probability of

i gray level of image. Select the optimal t ; make the image entropy

$$H(t) = -\sum_{i=0}^t \frac{P_i}{P_t} \ln \frac{P_i}{P_t} - \sum_{i=t+1}^{l-1} \frac{P_i}{1-P_t} \ln \frac{P_i}{1-P_t} \text{ be maximal. Extending to multi-}$$

threshold, the problem is: select multi-threshold t_k , make the image entro-

$$\text{py } H(t_1, t_2, \dots, t_k) = -\sum_{i=0}^{t_1} \frac{P_i}{P_{t_0}} \ln \frac{P_i}{P_{t_0}} - \sum_{i=t_1+1}^{t_2} \frac{P_i}{P_{t_1}} \ln \frac{P_i}{P_{t_1}} - \dots - \sum_{i=t_{k-1}}^{l-1} \frac{P_i}{P_{t_k}} \ln \frac{P_i}{P_{t_k}} \text{ be maximal.}$$

We optimize and select 2, 3 thresholds with improved AFSA for image segmentation.

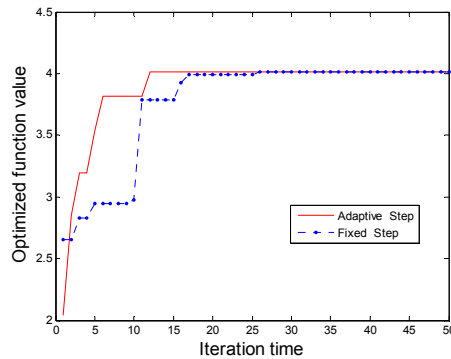


Fig. 9.1. Convergence speed with adaptive step and fixed step

9.3.2 The parameter selection and results

The parameter selection: fish number is 15, *try_number* is 3, iteration is 50, *Step* is 10, *Visual* is 10, crowded factor δ is 0.8 and the result is shown in the Figure 9.2.



(a) Source image (b). Single threshold (c) Two thresholds (d) Three thresholds

Fig. 9.2. The image segmentation with 1, 2 and 3 thresholds respectively

We also use the GA and PSO for the same problem, the GA parameter selection: population size is 20, iteration number is 100, P_c is 0.8 and P_m is 0.05, the PSO parameter selection: the number of particles is 20, the iteration number is 100, $w_{\max}=0.9$, $w_{\min}=0.4$, learning factor $c_1=c_2=2$, the result is shown in Table 9.1.

9.4 Discussion and conclusion

9.4.1 Discussion

From the principle of the optimization algorithm, the AFSA is a neighborhood search, with the aid of heuristic search strategy, and has the ability of the global optimization. The AFSA has no special requirements to the object function and initial values, so the initial value can be set with stochastic values or fixed values, and other parameters can be set in a wide range. In brief, the algorithm has strong adaptability.

Table 9.1. The comparison of GA, PSO and improved AFSA

Method	2-threshold	ME	Time	3-threshold	ME	Time
GA	82,151	12.613	17.24s	58,115,162	15.571	20.91s
PSO	80,151	12.613	8.67s	59,109,162	15.590	9.96s
AFSA	80,151	12.613	4.19s	60,109,162	15.590	6.9s

In the Table 9.1, with the optimized threshold results of image segmentation by GA, PSO and improved AFSA, we can see that the properties of AFSA are better than PSO, and the properties of PSO are better than GA. But how to compare them fairly is a problem, the AFSA has low computing complexity, faster convergence speed and more precision, the future work is consideration of establishing the theory standard and adjusting method of other parameters such as adaptive *Visual*, crowded factor effect etc. Fish fast recognition and decision mechanism can be considered into the algorithm and some other improved ideas can be seen in [10,13].

9.4.2 Conclusion

We propose three adaptive step methods in the AFSA and analyze the properties of improved method, and then we use it in the multi-threshold image segmentation, and compare it with the GA and PSO methods. The results show that the AFSA can obtain better performance than the GA and PSO. The some improved methods and ideas are proposed, the applied performance can be better, the AFSA can obtain the optimal or sub-optimal results on theory, it is believed that the AFSA can be used in some other complicated optimization applications.

Acknowledgments. This work is supported by the National Natural Scientific Foundation of China (No. 60672036, No. 60672037), the Natural Science Foundation of Shandong Province of China (No. Y2006G06), and the Catalan Government (Generalitat de Catalunya, Spain) under grant SGR2005-00690.

References

1. Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley Longman
2. Eberhart R, Shi Y, Kennedy J (2001) Swarm intelligence. Morgan Kaufmann, San Francisco
3. Dorigo M, Stzle T (2004) Ant colony optimization. MIT Press, Cambridge
4. Pham DT, Ghanbarzadeh A, Koc E (2005) The bees algorithm. Technical Note, Manufacturing Engineering Centre, Cardiff University, UK
5. Li XL (2003) A new intelligent optimization-artificial fish swarm algorithm. Doctor thesis, Zhejiang University of Zhejiang, China
6. Jiang MY, Yuan DF (2005) Wavelet threshold optimization with artificial fish swarm algorithm. In: Proc. of IEEE International Conference on Neural Networks and Brain, Beijing China, pp 569 -572
7. Xiao JM, Zheng XM, Wang XH (2006) A modified artificial fish-swarm algorithm. In: Proc. of IEEE the 6th World Congress on Intelligent Control and Automation, Dalian China, pp 3456-3460
8. Zhang MF, Cheng S, Li FC (2006) Evolving neural network classifiers and feature subset using artificial fish swarm. In: Proc. of IEEE International Conference on Mechatronics and Automation, Luoyang China, pp 1598-1602
9. Shan XJ, Jiang MY (2006) The routing optimization based on improved artificial fish swarm algorithm. In: Proc. of IEEE the 6th World Congress on Intelligent Control and Automation, Dalian China, pp 3658-3662
10. Jiang MY, Wang Y, Pfletschinger S, Lagunas MA (2007) Optimal multiuser detection with artificial fish swarm algorithm. In: Proc. of International Conference on Intelligent Computing (ICIC 2007), CCIS 2, Springer-Verlag Berlin Heidelberg, pp 1084-1093
11. Yu Y, Tian YF, Yin ZF (2005) Multiuser detector based on adaptive artificial fish school algorithm. In: Proc. of IEEE International symposium on communications and information technologym, pp 1480-1484
12. Jiang MY, Wang Y, Rubio F (2007) Spread Spectrum Code Estimation by Artificial Fish Swarm Algorithm. In: Proc. of IEEE International Symposium on Intelligent Signal Processing (WISP'2007), Alcalá de Henares, Spain
13. Jiang MY, Yuan DF (2006) Artificial fish swarm algorithm and its applications. In: Proc. of International Conference on Sensing, Computing and Automation, Chongqing China, pp 1782-1787
14. Zhang YJ (2001) Image segmentation. Science press, Beijing

Chapter 10

Propagation of elastic-plastic waves in bars

Olga Martin

University “Politehnica” of Bucharest, 313, Splaiul Independentei,
ROMANIA
omartin_ro@yahoo.com

Abstract. In this paper we study the impact loading of semi-infinite prismatic bars, using a numerical algorithm based on the finite element method (FEM) and finite-differences method (FDM). A detailed analysis of the calculus errors is made and a theorem concerning to the approximations is proved. The numerical example confirms the efficiency of the proposed method.

Keywords. Dynamic plasticity, finite element method, finite-differences method, constitutive equation, errors

10.1 Introduction

The action of loads suddenly applied to a body is not propagated instantaneously but is transmitted from one particle to another in a wave-like manner. The strength problems for various machines and structures subjected to impact (or explosions) can be studied only if a clear understanding is gained of the laws of propagation of elastic-plastic deformation preceding failure. Also, dynamic problems may be of importance in analysis of high-speed technological metal-forming processes.

In the beginning, Kachanov [6], Rakhmatulin [11], Karman and Duwez [7] have been published the studies regarding the propagation of elastic-plastic compression (extension) waves in a bar. Then, Kubenko [8], Bajons [2], Cristescu [4] were studied various generalizations of these problems. Detailed references may be found in the books [4, 8].

10.2 Problem formulation

Let us consider the problem of waves propagation in a semi-infinite prismatic bar, whose axis coincide with the x - axis. We consider the following basic assumptions.

- During deformation the cross sections of the bar remain plane and normal to the x - axis.
- Deformations are small and hence, changes in the dimensions of a bar can be neglected.
- The inertia forces corresponding to the motion of particles of a bar in transverse directions (due to the contraction or expansion of the section) can be neglected.
- The effect of the rate of strain on the curve representing the relation between stress $\sigma = \sigma_x$ and $\varepsilon = \varepsilon_x$ can be neglected. In dynamic problem, the rates of strain are large and their effect on stress – strain curve may be appreciable. In this paper, we adopt a relation $\sigma = \sigma(\varepsilon)$ that is a first approximation and it refers to a certain average strain rate over the given range.

Let us consider a bar of the cross section A and the density is ρ . The displacement of the point x at the time t is denoted by $u(x, t)$ and the strain is

$$\varepsilon = \frac{\partial u(x, t)}{\partial x} = u_x \text{ and } \dot{u} = \frac{\partial u(x, t)}{\partial t}$$

Suppose that the bar is in the rest at $t = 0$ and for the end $x = 0$ some disturbances are given at time $t \geq 0$. For instance, a velocity

$$\dot{u}\big|_{x=0} = \frac{\partial u(x, t)}{\partial t}\bigg|_{x=0} = V(t), t \in [0, T].$$

The total energy at the time t is the sum

$$W_{total} = K + P \quad (10.1)$$

where

$$K = \frac{\rho A}{2} \int_0^L \dot{u}^2 dx \quad (10.2)$$

is the kinetic energy,

$$P = A \int_0^L W dx$$

is the potential energy and W is the strain energy of a unit volume.

The strain energy W has the following properties:

- the derivatives $\frac{d^2 W}{du_x^2}$ are positive and bounded;
- $\frac{d^2 W}{du_x^2}$ is proportionally with the square of wave velocity;
- $\frac{dW}{du_x} = \sigma(u_x)$, where $\sigma(u_x)$ is the stress relative to the references configuration (Piola-Kirchhoff stress).

Then, the principle of energy conservation:

$$\dot{W}_{total} = 0 \quad (10.3)$$

will lead to the equation of motion. Its solution belongs to the measurable set $D = H^0([0, L]) \times [0, T]$, where $H^0([0, L]) = L_2([0, L])$ is a Hilbert space with the scalar product

$$(u(t), s(t)) = \int_0^L u(x, t)s(x, t)dx \quad (10.4)$$

and the norm

$$\|u(t)\|^2 = \int_0^L u^2(x, t)dx \quad (10.5)$$

for any $t \in [0, T]$. Then our problem (10.3) becomes

$$\rho A(\ddot{u}, \dot{u}) + A(\sigma(u_x), \dot{u}_x) = 0 \quad (10.6)$$

with the condition at the end $x = 0$:

$$\dot{u}(0, t) = V(t), \quad \forall t \in [0, T] \quad (10.7)$$

In order to get the solution of the problem (10.6) - (10.7), we consider on the x axis a points system: $0 = x_0 < x_1 < \dots < x_N$, with the constant step $h = x_{k+1} - x_k$, $k = 0, 1, 2, \dots, N$.

We seek an approximate solution of the form

$$U(x, t) = \sum_{k=1}^N \alpha_k(t) \phi_k(x) \quad (10.8)$$

where $\phi_k(x)$ are the coordinate functions:

$$\phi_k(x) = \begin{cases} \frac{x - x_{k-1}}{h}, & x \in [x_{k-1}, x_k] \\ \frac{x_{k+1} - x}{h}, & x \in [x_k, x_{k+1}] \\ 0, & x \notin [x_{k-1}, x_{k+1}] \end{cases}$$

with $\phi_k(x_k) = 1$. Let us now consider a further points system for the interval $[0, T]$: $0 = t_0 < t_1 < \dots < T_R = T$, with the constant step $\tau = t_l - t_{l-1}$, $l = 1, 2, \dots, R$. Using the notations:

$$\alpha_k^l = \alpha_k(l\tau), u^l = u(x, l\tau) \quad (10.9)$$

we define the progressive and regressive differences

$$\nabla_t u^l = \frac{u^{l+1} - u^l}{\tau}, \quad \Delta_t u^l = \frac{u^l - u^{l-1}}{\tau} \quad (10.10)$$

Then, the numerical derivatives of first and second order with respect to t , become

$$\begin{aligned} \delta_t u^l &= \frac{1}{2} (\nabla_t + \Delta_t) u^l = \frac{u^{l+1} - u^{l-1}}{2\tau} \\ \delta_t^2 u^l &= \nabla_t \Delta_t u^l = \frac{u^{l+1} - 2u^l + u^{l-1}}{\tau^2} \end{aligned} \quad (10.11)$$

Replacing (10.9) and (10.11) in the equation (10.6), this becomes

$$A\rho(\delta_t^2 U^l, \delta_t U^l) + A(\sigma(U_x^l), \delta_t U_x^l) = 0 \quad (10.12)$$

where

$$\begin{aligned} U^l &= U(x, l\tau) = \sum_{k=1}^N \alpha_k^l \phi_k(x) \\ U_x^l &= \frac{\partial U(x, l\tau)}{\partial x} = \sum_{k=1}^N \alpha_k^l \frac{d\phi_k(x)}{dx} \\ \delta_t U_x^l &= \sum_{k=1}^N \delta_t \alpha_k^l \frac{d\phi_k(x)}{dx} \end{aligned} \quad (10.13)$$

We now define the mass matrix

$$m_{kl} = \rho A \int_0^L \phi_k(x) \phi_l(x) dx = \rho A (\phi_k(x), \phi_l(x)) \quad (10.14)$$

and rewrite the equation (10.12) in the form

$$\sum_{k,i=1}^N m_{ki} \delta_t^2 \alpha_k^l \delta_t \alpha_i^l + A \left(\sigma \left(\sum_{k=1}^N \alpha_k^l \frac{d\phi_k(x)}{dx} \right), \sum_{i=1}^N \delta_t \alpha_i^l \frac{d\phi_i(x)}{dx} \right) = 0$$

$l = 1, 2, \dots, R-1$.

Since $\delta_t \alpha_i^l$ are arbitrary, we obtain for a fixed t_l the following system with unknowns α_i^l

$$\sum_{k=1}^N m_{ki} \delta_t^2 \alpha_k^l + A \left(\sigma \left(\sum_{k=1}^N \alpha_k^l \frac{d\phi_k}{dx} \right), \frac{d\phi_i}{dx} \right) = 0, \quad i = 1, 2, \dots, N \quad (10.15)$$

The second term of the equation (10.15) defines the rigidity of the model. For instance, in the case of the linear theory we have

$$\sigma(u_x^l) = E u_x^l = E \sum_{k=1}^N \alpha_k^l \frac{d\phi_k(x)}{dx} \quad (10.16)$$

where E is Young's modulus and

$$AE \left(\sum_{k=1}^N \alpha_k^l \frac{d\phi_k}{dx}, \frac{d\phi_i}{dx} \right) = \sum_{k=1}^N \gamma_{ki} \alpha_k^l \quad (10.17)$$

Here γ_{ki} are the elements of the stiffness matrix from the finite element theory. In our case, the constitutive equation is nonlinear. Such equation will be presented in the numerical example.

10.3 Approximation study

Let us now consider the derivable function $\sigma(u_x)$ with $c_0^2 = \inf |\sigma'(u_x)|$, $c_1^2 = \sup |\sigma'(u_x)|$. Then, for two gradients u_x and w_x , where w_x there is in a neighborhood of u_x , we have

$$\sigma(u_x) = \sigma(w_x) \pm \sigma'(w_x + \theta(u_x - w_x))(u_x - w_x), \quad \theta \in (0,1) \quad (10.18)$$

This relation leads to the following inequalities

$$\begin{aligned} c_0^2(u_x - w_x, V_x) &\leq (\sigma(u_x) - \sigma(w_x), V_x) \leq \\ &\leq c_1^2(u_x - w_x, V_x), \forall V \in D. \end{aligned} \quad (10.19)$$

In view of the finite differences theory, if $u(x, t)$ has the derivatives of the fourth order with respect to t bounded on D , we have for $t_l = l\tau$ that

$$\delta_t^2 u(x, l\tau) = \frac{\partial^2 u(x, l\tau)}{\partial t^2} + \omega^l(x, t) \cdot \tau \quad (10.20)$$

Let us now consider the exact solution u of the problem (10.6)-(10.7) and the numerical solution U defined by (10.7). By virtue of (10.12) and (10.20), we get

$$\rho(\delta_t^2 U^l, V) + (\sigma(U_x^l), V_x) = 0 \quad (a) \quad (10.21)$$

$$\rho(\delta_t^2 u^l, V) + (\sigma(u_x^l), V_x) = \rho(\omega^l \tau, V) \quad (b)$$

where $V = \delta_t U^l$. We now introduce an interpolation polynomial of the solution u for a fixed t_l

$$\tilde{u}^l = \tilde{u}(x_k, t_l) = u(x_k, t_l) = u^l, \quad k = 0, 1, \dots, N \quad (10.22)$$

and its interpolation error

$$\tilde{e}^l = \tilde{u}^l - u^l \quad (10.23)$$

We define for any fixed t_l

$$\varepsilon^l = \tilde{u}^l - U^l \quad (10.24)$$

and therefore, the finite element error is of the form

$$e^l = u^l - U^l = \varepsilon^l - \tilde{e}^l \quad (10.25)$$

Using the interpolation theorem of the finite element theory for the functions, which belong to the C^k functions space, [12], we remind the following properties: there are the constants μ and μ_1 independently of step h , such that

$$\|\tilde{e}^l\| \leq \mu h^2, \quad \|\tilde{e}_x^l\| \leq \mu_1 h \quad (10.26)$$

Lemma

Taking the relations (10.18) - (10.21) into account, there are for any t_l , the constant μ , independently of h , such that

$$c_0^2 \|\varepsilon_x^l\|^2 + \rho(\delta_t^2 e^l, \varepsilon^l) \leq \frac{c_1^2}{4\mu} \|\tilde{e}_x^l\|^2 + c_1^2 \mu \|\varepsilon_x^l\|^2 + \rho(\omega^l \tau, \varepsilon^l) \quad (10.27)$$

Let us consider (10.21a), where u^l is replaced by $\tilde{u}^l$ and $V = \varepsilon^l = \tilde{u}^l - U^l$

$$\rho(\delta_t^2 \tilde{u}^l, \tilde{u}^l - U^l) + (\sigma(\tilde{u}_x^l), \tilde{u}_x^l - U_x^l) = 0 \quad (10.28)$$

Subtracting (10.28) and (10.21a), where $V = \tilde{u}^l - U^l$ and using (10.19), we get

$$\begin{aligned} & \rho(\delta_t^2 (\tilde{u}^l - U^l), \tilde{u}^l - U^l) + c_0^2 \|\tilde{u}_x^l - U_x^l\|^2 \leq \\ & \leq \rho(\delta_t^2 (\tilde{u}^l - U^l), \tilde{u}^l - U^l) + (\sigma(\tilde{u}_x^l) - \sigma(U_x^l), \tilde{u}_x^l - U_x^l) \leq \\ & \leq \rho(\delta_t^2 (\tilde{u}^l - U^l), \tilde{u}^l - U^l) + c_1^2 \|\tilde{u}_x^l - U_x^l\|^2 \end{aligned} \quad (10.29)$$

With the help of the inequality

$$(a, b) \leq \frac{1}{4\mu} \|a\|^2 + \mu \|b\|^2, \quad \mu > 0 \quad (10.30)$$

we obtain by the addition and subtraction of u^l in the first term and of the stress $\sigma(u_x^l)$ in the second term the following inequalities

$$\begin{aligned} & \rho(\delta_t^2 \varepsilon^l, \varepsilon^l) + c_0^2 \|\varepsilon_x^l\|^2 = \rho(\delta_t^2 (\tilde{u}^l - U^l), \tilde{u}^l - U^l) + c_0^2 \|\tilde{u}_x^l - U_x^l\|^2 \leq \\ & \leq \rho(\delta_t^2 (\tilde{u}^l - u^l + u^l - U^l), \tilde{u}^l - U^l) + \\ & \quad + (\sigma(\tilde{u}_x^l) - \sigma(u_x^l) + \sigma(u_x^l) - \sigma(U_x^l), \tilde{u}_x^l - U_x^l) \leq \\ & \leq \rho(\delta_t^2 (\tilde{u}^l - u^l), \tilde{u}^l - U^l) + (\sigma(\tilde{u}_x^l) - \sigma(u_x^l), \tilde{u}_x^l - U_x^l) + \\ & \quad + \rho(\omega^l \tau, \tilde{u}^l - U^l) \leq \rho(\delta_t^2 \tilde{e}^l, \varepsilon^l) + c_1^2 (\tilde{e}_x^l, \varepsilon_x^l) + \rho(\omega^l \tau, \varepsilon^l) \leq \\ & \leq \rho(\delta_t^2 \tilde{e}^l, \varepsilon^l) + \frac{c_1^2}{4\mu} \|\tilde{e}_x^l\|^2 + c_1^2 \mu \|\varepsilon_x^l\|^2 + \end{aligned} \quad (10.31)$$

$$+ \rho(\omega^l \tau, \varepsilon^l) \leq \rho(\delta_t^2 \tilde{e}^l, \varepsilon^l) + \frac{c_1^2}{4\mu} \|\tilde{e}_x^l\|^2 + c_1^2 \mu \|\varepsilon_x^l\|^2 + \rho(\omega^l \tau, \varepsilon^l)$$

Finally, we obtain

$$c_0^2 \|\varepsilon_x^l\|^2 + \rho(\delta_t^2 e^l, \varepsilon^l) \leq \frac{c_1^2}{4\mu} \|\tilde{e}_x^l\|^2 + c_1^2 \mu \|\varepsilon_x^l\|^2 + \rho(\omega^l \tau, \varepsilon^l) \quad (10.32)$$

Theorem

If (10.27) exists for a fixed t_l , then the finite element error verifies the inequality

$$\|e^l\| \leq Ch \quad (10.33)$$

Subtracting the relations (10.21), where $V = \varepsilon^l$ and using (10.19) and (10.30), we get

$$\begin{aligned} \rho(\delta_t^2 e^l, \varepsilon^l) &\leq (\omega^l \tau, \varepsilon^l) - c_0^2 (\varepsilon_x - \tilde{e}_x^l, \varepsilon_x^l) \leq \\ &\leq (\omega^l \tau, \varepsilon^l) + c_0^2 (\tilde{e}_x^l, \varepsilon_x^l) \leq (\omega^l \tau, \varepsilon^l) + \frac{c_0^2}{4\mu} \|\tilde{e}_x^l\|^2 + c_0^2 \mu \|\varepsilon^l\|^2 \end{aligned} \quad (10.34)$$

Subtracting (10.27) and (10.34), we have

$$c_0^2 \|\varepsilon_x^l\|^2 \leq \frac{c_1^2 - c_0^2}{4\mu} \|\tilde{e}_x^l\|^2 + (c_1^2 - c_0^2) \mu \|\varepsilon_x^l\|^2$$

Consequently,

$$(c_0^2 - c_1^2 \mu + c_0^2 \mu) \|\varepsilon_x^l\|^2 \leq ah^2$$

If $\mu \in (0, 1)$, we obtain

$$\|\varepsilon_x^l\|^2 \leq C_1 h^2 \quad (10.35)$$

Using the definition (10.25) of e^l , we get

$$\|e_x^l\| \leq \|\varepsilon_x^l\| + \|\tilde{e}_x^l\| \leq C_2 h + \mu_1 h = C_3 h$$

and in accordance with the Friedrichs's inequality, we finally have

$$\|e^l\| \leq k \|e_x^l\| \leq Ch \quad (10.36)$$

10.4 Numerical example

We study the waves propagation in a semi-infinite bar $x \geq 0$, which is at rest at $t = 0$. Let us consider that at $t > 0$ the velocity V of the point $x = 0$ is of the form

$$\dot{U}(0, t) = \begin{cases} 2V(t/t_0), & t < t_0/2 \\ V, & t \geq t_0/2 \end{cases}, \quad V = 2000 \text{ cm/sec}, t_0 = 5 \cdot 10^{-6} \text{ sec} \quad (10.37)$$

The density $\rho = 0.00785 \text{ kg/cm}^3$ and the constitutive equation is of the form, [7]:

$$\sigma = \sigma_c + K \cdot \left(|\varepsilon|^{0.4} - |\varepsilon_c|^{0.4} \right) \quad (10.38)$$

where $\sigma_c = 500 \text{ daN/cm}^2$, $E = 10^6 \text{ daN/cm}^2$, $K = 5000 \text{ daN/cm}^2$.

The step $h = 0.05 \text{ cm}$ and the time step $\tau = h / c_0 = 4.43 \cdot 10^{-6}$, where $c_0 = \sqrt{E / \rho}$ is the velocity of the elastic wave and $\varepsilon_c = \sigma_c / E$.

The displacements U of the bar points will be calculated by finite element - finite differences methods for the time intervals $[i\tau, (i+1)\tau]$, $i = 0, 1, 2, \dots$

Step 1. $0 \leq t \leq \tau$

$$U_{11} = U(x_1, \tau) = V \tau^2 / t_0 = 7.85 \cdot 10^{-3} \text{ cm}$$

From the definition (10.7), we observe that

$$U(x_k, t_l) = \alpha_k^l \Rightarrow U(x_1, t_1) = \alpha_1^1 \quad (10.39)$$

Step 2. $\tau \leq t \leq 2\tau$

$$U_{12} = \alpha_1^2 = U(x_1, 2\tau) = V \cdot (2\tau)$$

$$U_{22} = \alpha_2^2 = U(x_2, 2\tau) = U_{11}, \quad x_2 = 2h$$

Evidently, the displacements $U_{ik} = 0$, if $k < i$, this assumption been in good agreement with the experimental observations.

Step 3. $2\tau \leq t \leq 3\tau$

$$U_{33} = U_{11} = \alpha_3^3, \quad U_{13} = \alpha_1^3 = V(3\tau)$$

By using the system (10.15) and the definition (10.12), we can obtain $U(x_2, t_3) = \alpha_2^3$ from the equation

$$\begin{aligned} \rho \left[\delta_t^2 \alpha_1^2 \int_{x_1}^{x_2} \phi_2 \phi_1 dx + \delta_t^2 \alpha_2^2 \int_{x_1}^{x_3} \phi_2 \phi_2 dx + \delta_t^2 \alpha_3^2 \int_{x_2}^{x_3} \phi_2 \phi_3 dx \right] = \\ = -K \left| \frac{U_{22} - U_{12}}{h} \right|^{0.4} + K \left| \frac{-U_{22}}{h} \right|^{0.4} \end{aligned} \quad (10.40)$$

Hence we find $\alpha_2^3 = U_{23}$.

Step 4. $3\tau \leq t \leq 4\tau$

$$U_{14} = \alpha_1^4 = V \cdot (4\tau) \quad \text{and} \quad U_{44} = \alpha_4^4 = U_{11}$$

The system (10.15) becomes

$$\rho \left[\delta_t^2 \alpha_1^3 \int_{x_1}^{x_2} \phi_2 \phi_1 dx + \delta_t^2 \alpha_2^3 \int_{x_1}^{x_3} \phi_2 \phi_2 dx + \delta_t^2 \alpha_3^3 \int_{x_2}^{x_3} \phi_2 \phi_3 dx \right] =$$

$$= - \left(\sigma \left(\sum_{k=1}^3 \alpha_k^3 \frac{d\phi_k}{dx} \right), \frac{d\phi_2}{dx} \right)$$

where, for instance

$$\left(\sigma \left(\sum_{k=1}^3 \alpha_k^3 \frac{d\phi_k}{dx} \right), \frac{d\phi_2}{dx} \right) = 5000 \left[\left| \frac{U_{23} - U_{13}}{h} \right|^{0.4} - \left| \frac{U_{33} - U_{23}}{h} \right|^{0.4} \right]$$

Finally, we find U_{24} and U_{34} from the iterative matrix equation

$$\begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} U_{24} \\ U_{34} \end{bmatrix} - 2 \begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} U_{23} \\ U_{33} \end{bmatrix} + \begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} U_{22} \\ 0 \end{bmatrix} =$$

$$= - \begin{bmatrix} \left(\sigma \left(\sum_{k=1}^3 \alpha_k^3 \frac{d\phi_k}{dx} \right), \frac{d\phi_2}{dx} \right) + U_{14} - 2U_{13} + U_{12} \\ \left(\sigma \left(\sum_{k=1}^3 \alpha_k^3 \frac{d\phi_k}{dx} \right), \frac{d\phi_3}{dx} \right) + U_{44} \end{bmatrix}$$

or

$$D \cdot U^{(4)} = -B + 2D \cdot U^{(3)} - D \cdot U^{(2)} \quad (10.41)$$

where

$$D = \begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}$$

Step $n+1$. $n\tau \leq t \leq (n+1)\tau$

The matrix equation (10.41) becomes

$$D \cdot U^{(n+1)} = -B + 2D \cdot U^{(n)} - D \cdot U^{(n-1)} \quad (10.42)$$

where

$$D = \begin{bmatrix} 4 & 1 & 0 & 0 & \cdots & 0 \\ 1 & 4 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 4 & 1 \\ 0 & 0 & \cdots & 0 & 1 & 4 \end{bmatrix}.$$

The values of $U_{k,l} = U(x_k, t_l)$, for $k = 0, 1, \dots, 4$ and $l = 0, 1, \dots, 4$, are

$$U = \begin{bmatrix} 0.00785 & 0.018 & 0.027 & 0.035 & 0.044 \\ 0 & 0.00785 & 0.018 & 0.027 & 0.035 \\ 0 & 0 & 0.00785 & 0.014 & 0.02 \\ 0 & 0 & 0 & 0.00785 & 0.014 \\ 0 & 0 & 0 & 0 & 0.00785 \end{bmatrix}$$

In the Figure 10.1 are presented the values of the displacements $U(x_i, t_l)$, that correspond to the sections $x = x_k$, $k = 1, 2, \dots, 5$ for the time: t_l , $l = 1, 2, 3, 4$.

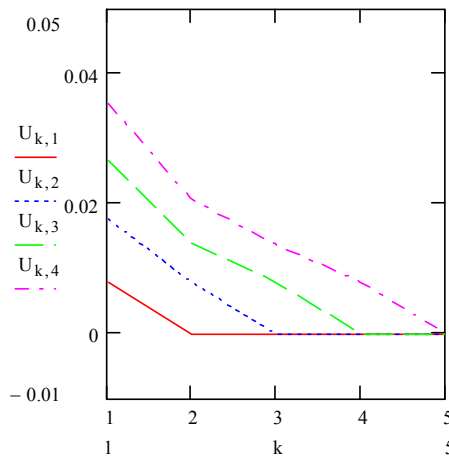


Fig. 10.1. Values of the displacements $U(x_i, t_l)$

10.5 Conclusions

The action of loads suddenly applied to a structure is not propagated instantaneously but it is transmitted from one particle to another in a wave-like manner. In the numerical example we approach the case of large plastic strains, disregarding the elastic strains. In this case, the waves velocity is variable, the larger strain propagating with a lower velocity; as time goes on, the waves “diffuses”. Unlike the characteristics method, the effects of this phenomenon and the interactions of the direct and reflected waves are automatically determined with the help of the numerical method presented in the paper.

The study of the errors that correspond to the numerical solution shows that these are of the first order with respect to the spatial step

h. In a future paper we will apply the finite element-finite differences method to the unloading phenomenon of the elastic-plastic bars.

References

1. Babuska I, Szabo B (1991) Finite element analysis. John Wiley and Sons New York
2. Bajons P (1980) The problem of resonance for longitudinal small amplitude excitation of an elastic-plastic bar. *Acta Mecanica* 37:199-215
3. Benitez F, Lu L, Rosakis A (1993) A boundary element formulation based on the three-dimensional elasto-static fundamental solution. *International Journal for Numerical Methods in Engineering* 36:3097-3130
4. Cristescu N (2007) Dynamic plasticity. World Scientific
5. Hongwei Ma (1999) Experimental studies on dynamic plastic buckling of circular cylindrical shells under axial impact. *Acta Mecanica* 15:275-282
6. Kachanov L (2004) Fundamentals of the theory of plasticity. Dover Publications
7. Karman T, Duwez P (1950) The propagation of plastic deformation in solids. *Journal of Applied Physics* 21:987-994
8. Kubenko V (1972) Symposium on the propagation of elastic-plastic waves. *International Applied Mechanics* 8(5)
9. Mastorakis N, Martin O (2007) Finite Elements. WSEAS Press
10. Proudhomme S, Oden J, Westermann T, Bass J, Botkin M (2003) Practical methods for a posteriori error estimation in engineering applications. *International Journal for Numerical Methods in Engineering* 56:1193-1199
11. Rakhmatulin K, Demianov Y (1961) Strength under Intense short-time loads. MIR Publishers, Moscow
12. Reddy D, Oden J (1976) An introduction to mathematical theory of MEF. J. Wiley and Sons
13. Schreyer H (2005) Consistent diagonal mass matrices and finite element equations for one-dimensional problems. *International Journal for Numerical Methods in Engineering* 12:1171-1184
14. Voyiadjis G, Palazotto A, Gao X (2002) Modeling of metallic materials at high strain rates with continuum damage mechanics. *Applied Mechanics Reviews* 55:481-493

Chapter 11

Shahyat algorithm as a clustering method

Omid Khayat, Hamid Reza Shahdoosti

Department of Biomedical Engineering, Amirkabir University, IRAN

Abstract. This paper presents a new method of clustering and classification. Here is discussed a new method of clustering based on defining the heat function. Shahyat algorithm can be implemented either in supervised or unsupervised manner. We consider each pattern as a heat source; points of feature space are affected by the heat of all patterns. Therefore, the closer the patterns, the higher the heat, and we can determine compact regions of feature space noticing the heat of space. We calculate the heat of patterns, and choose cluster centers based on their heat. Finally, we solve the iris clustering problem and present several specifications of shahyat algorithm.

Keywords. Clustering, classification, Euclidean distance, heat function

11.1 Introduction

There are several methods of pattern clustering and classification. Clustering and classification are among the most important topics in artificial intelligence, signal processing, image processing, pattern recognition, etc.

Several popular approaches are known for clustering and classification such as k-means [6], nearest neighbor [5] and fuzzy c-means [2].

Shahyat algorithm presents a new method for clustering and classification. We have a large number of patterns and want to cluster them into a identical number. Shahyat algorithm is based on definition of a Heat Function. We consider each pattern in the N-dimensional feature space as a heat

source. Each pattern affects the space totally, that makes every point of N-dimensional feature space be affected by all patterns. This definition helps us to discern the concentration of patterns.

Because of high concentration in clusters, heat in the clusters is higher than out of clusters, and this fact is used to determine and confine the clusters. By calculating the heat of patterns, we can find cluster centers. Each pattern belongs to nearest center. Therefore the clusters are being determined.

11.2 Description of Shahyat algorithm

We consider k patterns in the N-dimensional feature space as below:

$$\begin{aligned} pattern &= \{X_1, X_2, \dots, X_k\} \\ X_i &= (d_1, d_2, \dots, d_n) (i=1, 2, \dots, k) \end{aligned} \quad (11.1)$$

In which X_i ($i=1, 2, \dots, k$) is the i 'th pattern and d_j ($j=1, 2, \dots, k$) is the j 'th feature of patterns. We want to classify k patterns into C clusters. First, we map each pattern as a point in the N-dimensional feature space. Assume that the N-dimensional space is orthogonal. The difference between two patterns is the Euclidean distance between them. We want to assign patterns with lower distance to the same cluster; since the patterns with low distance cause concentration, we need to determine high and low concentration of patterns in the space. We consider each pattern as a heat source to determine concentration by heat effectiveness. Patterns (heat sources) affect the whole space. It is assumed that the interaction of near points (in a cluster) is considerable. On the other hand it is negligible when the points are far from each other. This means that the effect decreases when the distance increases. Heat function is defined as below:

$$H(X_0|X_j) = F(DISTANCE \text{ BETWEEN } X_0 \& X_j) (j=1, 2, \dots, k) \quad (11.2)$$

where $H(X_0|X_j)$ is the heat of arbitrary point X_0 (which is affected by pattern X_j), which depends on the distance between X_0 and pattern X_j . F is a function of distance that should satisfy 3 conditions:

1. $F(0) < \infty$
2. $F(\infty) \rightarrow 0$
3. if $d < d' \rightarrow F(d) > F(d')$

The first condition points to the heat limitation in patterns, the second condition expresses that the heat is negligible at far, and the third shows that the function is decreasing by distance.

By increasing the distance from the patterns, heat decreases by a decreasing function (for example exponentially function). It is obvious that intensity of heat could be explanatory of high or low concentration of patterns. The heat can be calculated for each arbitrary point X_0 affected by K patterns as below:

$$H(X_0) = \sum_{j=1}^k H(X_0|X_j) \quad (11.3)$$

Here, $H(X_0|X_j)$ is the heat of point X_0 affected by pattern X_j and $H(X_0)$ is the heat of arbitrary point X_0 totally (affected by all patterns). First, it is needed to define a decreasing function and use it as F .

We can determine the decreasing function F by considering that the effect of each pattern on patterns in other clusters should be negligible. By having the necessary decrease in each direction we can determine the type of function (exponential or any other nonlinear decreasing function) and the coefficients in the function. For example when clusters are close, we consider F high decreasing function to have low heat between clusters (relative to in the clusters).

After determining the decreasing function F , we calculate the heat of all patterns. Then sort them according to their heat.

Let the sorted collection be $\{X_1, X_2, \dots, X_k\}$ where X_1 has the maximum heat and X_k has the minimum heat among the pattern collection.

Now we determine the straight line connecting patterns X_j and X_{j+1} ($j=1, 2, \dots, k-1$), and calculate the Heat function on points on the path from X_1 to X_k . Then we plot the heat as a function of the patterns met on the path.

Now we have a continuous diagram (if we define F to be a continuous) or a discrete diagram (if we define F to be a discrete function). By traversing the path of X_1 to X_k , we move in the clusters and between clusters. Out of clusters, the heat is low. Therefore it's needed to find the minimums of heat function (on the path). The minimums should satisfy a condition as below:

$$\frac{dH(l)}{dl} = 0 \rightarrow \quad (11.4)$$

$$l = l_1, l_2, \dots, l_T \left[(H(X_{previous}) - H(l_i)) \geq H(X_k) \right]_{(i=1,2,\dots,T)}$$

Whereby $\frac{dH(l)}{dl}$ is the differential of heat function relative to the path, $X_{previous}$ is l_i 's previous pattern, and $l_i (i = 1, 2, \dots, T)$ are the roots of the derivative.

If the heat difference between minimum and previous pattern is higher than $H(X_k)$, the first pattern after minimum (on the path) is being considered as a candidate for the center of the cluster (assumed that minimums occur out of clusters).

Furthermore, we can find the candidate centers without calculating the minima of function (without differentiating the Heat function relative to path) as below.

11.2.1 Finding the candidate centers without differentiating from Heat function

The heat of points in the whole space can be calculated. We traverse the path of X_1 to X_k (after X_j , we go to X_{j+1}), considering the heat of path, where the gradient of heat on a line (the line between two patterns) becomes higher than $H(X_k)$, first next pattern is a candidate center. Because it is first pattern after a minimum. By traversing the path of X_1 to X_k , we find all patterns that can be a candidate center.

Assumed that we have T candidate centers and one main center X_1 , totally $T + 1$ candidate center. Considering the C centers are needed, 3 cases may occur:

- First case: $C > T + 1$, it means that the number of candidate centers ($T + 1$) is less than the needed number of centers C , so, we should select $C - (T + 1)$ minima among other minimum. Now we extenuate $H(X_k)$ (it results the number considered center increases) till the number of candidate center be equal to or larger than needed center. Therefore we have $C \leq T + 1$.

- Second case: $C = T + 1$, it means that needed number of centers is equal to the number of candidate centers, so we choose the $T + 1$ candidate centers as the main centers.
- Third case: $C < T + 1$, this case often occurs, it means the number of candidate centers is more than the number of needed centers. We should select C center from $T + 1$ candidate centers by the following method.

11.2.2 Selecting C main centers from $T + 1$ candidate centers

First we calculate the distance between every two centers among $T + 1$ candidate centers. Therefore, we have $\binom{T+1}{2}$ calculated distances. Assume that the shortest calculated distance (they belong to the same cluster) is between X_m and X_n , and $H(X_m) \geq H(X_n)$, Shahyat algorithm removes X_n from the candidate center list (it puts X_n in the cluster with X_m as the center).

Now we have T candidate centers, we find the shortest distance (after removing X_n) and assume that the shortest distance is between X_p and X_q , ($H(X_p) \geq H(X_q)$) therefore remove the X_q . This procedure resume until the number of candidate centers becomes equal to the needed number of centers.

When the number of candidate centers becomes equal to C , each pattern belongs to the nearest center. The clustering is finished and the clusters are determined.

11.3 Solving a clustering problem by Shahyat algorithm

We have 40 patterns $X_j = (d_1, d_2)_{(j=1,2,\dots,40)}$ in a 2-dimensional feature space (d_1, d_2) , and want to assign them to 3 clusters. The patterns are shown in Figure 11.1, in which the horizontal axis represents feature d_1 and vertical axis represents feature d_2 . Now we calculate the heat of patterns based on Shahyat algorithm. Assumed that our arbitrary decreasing function is an exponentially decreasing function, as below:

$$H(X_0) = \sum_{j=1}^k e^{-Func(\|X_0 - X_j\|)} \quad (11.5)$$

where $H(X_0)$ is the heat of the arbitrary point X_0 affected by 40 pattern $\{X_1, X_2, \dots, X_{40}\}$, and $Func(\|X_0 - X_j\|)$ is the function that correlates with the distance between point X_0 and point X_j in the feature space. We consider $Func(\|X_0 - X_j\|)$ as below:

$$Func(\|X_0 - X_j\|) = \mu_1 (d_{1X_0} - d_{1X_j})^2 + \mu_2 (d_{2X_0} - d_{2X_j})^2 \quad (11.6)$$

where $X_0 = (d_{1X_0}, d_{2X_0})$, $X_j = (d_{1X_j}, d_{2X_j})$ and μ_1, μ_2 are the coefficients that determine the scale of the heat decrease in each direction. Noticing the distance between patterns in a cluster and between clusters at each direction we assume $\mu_1 = \mu_2 = 1$ (the problem give us this information).

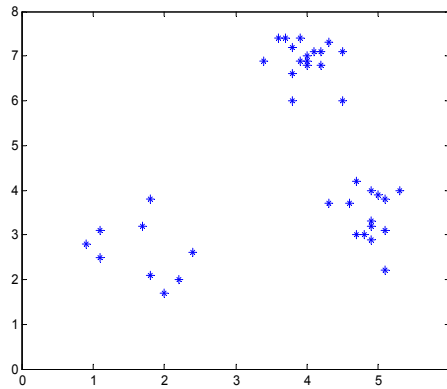


Fig. 11.1. Separation of 40 patterns in the 2-dimensional feature space

So the Heat function for arbitrary point X_0 is:

$$H(X_0) = \sum_{j=1}^k e^{-\left[(d_{1X_0} - d_{1X_j})^2 + (d_{2X_0} - d_{2X_j})^2 \right]} \quad (11.7)$$

According to Shahyat algorithm, we calculate the heat of patterns and then sort them from maximum to minimum. The diagram of Heat function

versus path (from X_1 with maximum heat to X_{40} with minimum heat) is shown in Figure 11.2.

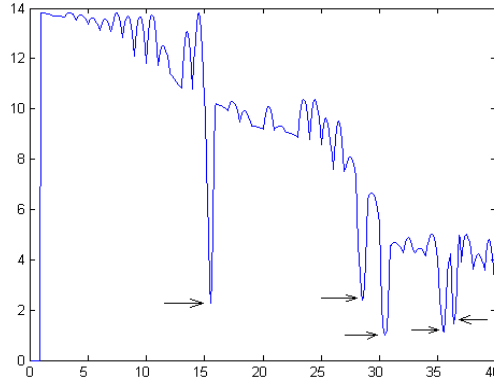


Fig. 11.2. The diagram of Heat function and minimums of function

The diagram has 5 minima that heat difference between the minimum and previous pattern is higher than $H(X_k)$. In Figure 11.3 the first pattern after each minimum is determined.

The candidate center collection is $\{X_1, X_{16}, X_{29}, X_{31}, X_{36}, X_{37}\}$.

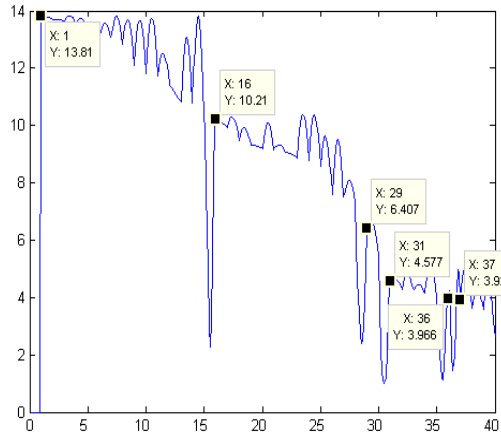


Fig. 11.3. First pattern after each minimum is determined

The shortest distance is the distance between X_1 and X_{29} (the distance is 1.04) so Shahyat algorithm removes X_{29} from candidate center collection (because $H(X_1) \geq H(X_{29})$), after removing X_{29} from the candidate

center collection, the shortest distance is between X_{16} and X_{36} (it is 1.25).

Hence, the algorithm removes X_{36} from candidate center collection (because $H(X_{16}) \geq H(X_{36})$).

Now we have 4 centers and need to remove one center, the shortest distance is between X_{31} and X_{37} , so we remove X_{37} (because $H(X_{31}) \geq H(X_{37})$). Now we have 3 centers, therefore, we consider them as the main centers. The centers are shown in Figure 11.4.

Now the centers are determined. Each pattern belongs to the nearest center. Now clusters are specific.

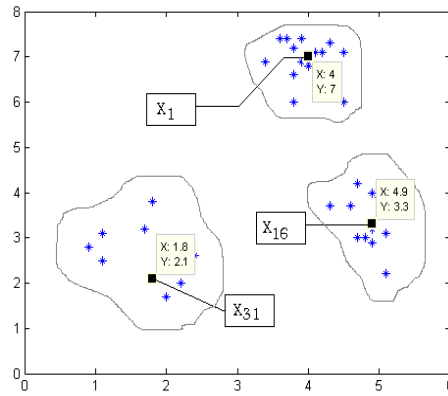


Fig. 11.4. Clustered pattern

11.4 Specifications of Shahyat algorithm

1. The algorithm is complete; this means that under each condition the algorithm can cluster and does not depend on special conditions. Because of in the clusters, the patterns are near together, and concentration in the clusters is higher than out of clusters, hence the heat in the clusters is more than out of clusters. So by defining an appropriate heat function the algorithm (based on heat in the clusters and out of clusters) can separate the clusters.
2. The maximum complexity of solving each problem by Shahyat algorithm is known, which depends on number of patterns, number of cluster that we want, distribution of patterns in the feature space and dimension of the patterns. It seems Shahyat algorithm after choosing

candidate centers, works like nearest neighbor. But solving the classification problems by Shahyat algorithm has less complexity, necessary memory and needed processing relative to solving by k-means, nearest neighbor, fuzzy c-means.

3. By running the algorithm once, we can classify the patterns into an arbitrary number of clusters, and it is not needed to run the algorithm again when we want to change the number of clusters. But k-means should process the algorithm procedures again if the number of clusters (that we want) changes.
4. For a certain problem, the algorithm has the ability to reduce the error (if error exists) by changing the type of Heat function and the coefficients in function. For example by varying the coefficients μ_1, μ_2 in exponential function (depend on the distance between patterns in a cluster and the distance between clusters), the classification rate will change.
5. Shahyat algorithm is an informed method for clustering and classification. It uses the specifications of concentration, whereas k-means, nearest neighbor, fuzzy c-means are uninformed methods (based on Euclidean distance).

11.5 Iris data

To show that our algorithm also work fine for real data sets, we use the famous iris plant data set. The data set contains 3 classes, where each class refers to a type of plants. Iris data are four dimensional; we show it in every two dimensions (Figure 11.6).

Now we consider segments 1 and 2 in Figure 11.6.

We cluster them by Shahyat algorithm. Assume that the decreasing function F of Shahyat algorithm is exponential with coefficients μ_1, μ_2 . The diagram of clustering error relative to gradient of μ for two segments is shown in Figure 11.5.

As we see, the error of clustering for segment 1 is lower than segment 2 (when we consider it only 2-dimensional). By considering iris data in 4-dimensional, Shahyat algorithm cluster iris data with classification rate 96%.

Most works report a classification rate that does not exceed 90%. K-means and fuzzy c-means, for example both have a classification rate of 89.3% on the iris data set. Alternative fuzzy c-means (AFCM) (Wu and Yang, 2002), reaches a classification rate of 91.3%.

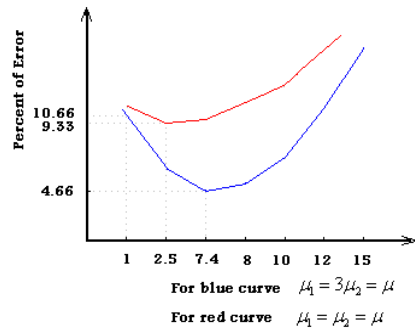


Fig. 11.5. Percent of error relative to variation of μ_1, μ_2 , segment 1 (blue curve) and segment 2 (red curve)

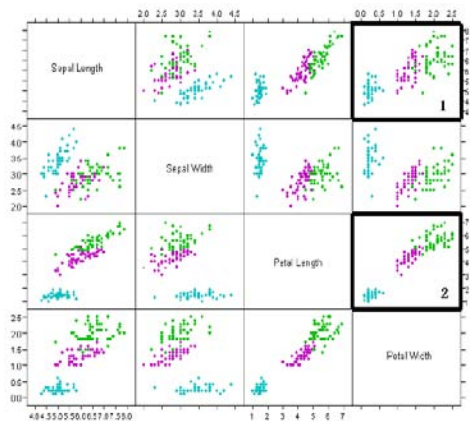


Fig. 11.6. Iris data

The block diagram of classification rate is shown in Figure 11.7.

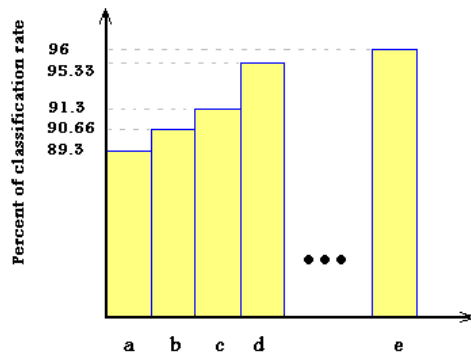


Fig. 11.7. Block diagram of classification rate for iris data clustering

where:

- a: k-means & fuzzy c-means
- b: segment 2 in figure 6 by Shahyat algorithm
- c: AFCM
- d: segment 1 in figure 6 by Shahyat algorithm
- e: Shahyat algorithm (4-dimensional)

11.6 Conclusion

Introduced algorithm, named Shahyat algorithm, makes use of the specifications of concentration by defining the heat function. This means that by defining an appropriate heat function, we have larger classification rate compared to methods based on Euclidean distance. Hence, Shahyat algorithm has higher classification rate compared to uninformed methods that only are based on Euclidean distance.

References

1. Wallace Manolis, Kollias Stefanos (2004) Robust, generalized, quick and efficient agglomerative clustering.
2. Alanzado Arnold C (2003) Noise clustering using fuzzy c-means and mixture distribution model.
3. Grabusts Peter, Borisov Arkady (2002) Using grid-clustering methods in data classification.
4. Roze Kenneth (1998) Deterministic annealing for clustering compression, classification, regression, and related optimization problems.
5. Eick Christoph F, Zeidat Nidal, Vilalta Ricardo (2005) Using representative-based clustering for nearest neighbor dataset editing.
6. Kanungo Tapas, Mount David M, Netanyahu Nathan S, Piatko Christine D (2003) A local search approximation algorithm for k-means clustering.
7. Ding Chris, He Xiaofeng (2004) K-nearest-neighbor consistency in data clustering: incorporating local information into global optimization.
8. Osipova Nataly (2004) Classification and clustering methods development and implementation for unstructured documents collections. Department of Programming Technology, Faculty of Applied Mathematics and Control Processes, St. Petersburg State University

Chapter 12

A modified electromagnetism-like algorithm based on a pattern search method

Ana Maria A. C. Rocha, Edite M. G. P. Fernandes

Minho University, Campus de Gualtar, 4710-057 Braga, Portugal
{arocha;emgpf}@dps.uminho.pt

Abstract. The Electromagnetism-like (EM) algorithm, developed by Birbil and Fang [2] is a population-based stochastic global optimization algorithm that uses an attraction-repulsion mechanism to move sample points towards optimality. A typical EM algorithm for solving continuous bound constrained optimization problems performs a local search in order to gather information for a point, in the population. Here, we propose a new local search procedure based on the original pattern search method of Hooke and Jeeves, which is simple to implement and does not require any derivative information. The proposed method is applied to different test problems from the literature and compared with the original EM algorithm.

Keywords. Global optimization, electromagnetism-like algorithm, pattern search method.

12.1 Introduction

Many real life global optimization problems that arise in areas such as physics, chemistry, and molecular biology, involve multi-modal and non-differentiable nonlinear functions of many variables that are difficult to handle by conventional gradient-based algorithms. As a result, many researchers have devoted themselves in finding reliable stochastic global optimization methods that do not require any derivative computation. Recent-

ly, Birbil and Fang proposed the electromagnetism-like (EM) algorithm that is a population-based stochastic search method for global optimization [1,2]. This algorithm simulates the electromagnetism theory of physics by considering each point in the population as an electrical charge. The method uses an attraction-repulsion mechanism to move a population of points towards optimality. The original algorithm incorporates a simple random local search procedure that is applied coordinate by coordinate to one point only or to all points in the population [2]. In an attempt to improve the accuracy of the results and to accelerate convergence we propose a modification to the original EM algorithm by replacing the random local search with a pattern search method [6] with guaranteed convergence.

The method is to work on nonlinear optimization problems with box constraints in the following form:

$$\min f(x), \quad \text{subject to } x \in \Omega \quad (12.1)$$

where $f: \mathcal{R}^n \rightarrow \mathcal{R}$ is a nonlinear function and $\Omega = \{x \in \mathcal{R}^n : l \leq x \leq u\}$ is a bounded feasible region.

The paper is organized as follows. Sect. 12.2 briefly introduces the original EM algorithm and Sect. 12.3 is devoted to describe the main ideas concerning the Hooke and Jeeves pattern search method. Sect. 12.4 contains the numerical results and some conclusions are drawn in Sect. 12.5.

12.2 Electromagnetism-like algorithm

The EM algorithm starts with a population of randomly generated points from the feasible region. Analogous to electromagnetism, each point is a charged particle that is released to the space. The charge of each point is related to the objective function value and determines the magnitude of attraction of the point over the population. The better the objective function value, the higher the magnitude of attraction. The charges are used to find a direction for each point to move in subsequent iterations.

The regions that have higher attraction will signal other points to move towards them. In addition, a repulsion mechanism is also introduced to explore new regions for even better solutions. The following notation is used: $x^i \in \mathcal{R}^n$ denotes the i th point of a population; x^{best} is the point that has the least objective function value; $x_k^i \in \mathcal{R}$ ($k = 1, \dots, n$), is the k th coordinate of the point x^i of the population; m is the number of points in the population; $MaxIt$ is the maximum number of EM iterations; $LSIt$ denotes the maxi-

num number of local search iterations; and δ is a local search parameter, $\delta \in [0, 1]$.

The EM algorithm comprises four main procedures.

Algorithm EM(m , MaxIt, LSIt, δ)

Initialize()

iteration $\leftarrow 1$

while termination criteria are not satisfied **do**

$Local(LSIt, \delta)$

$F \leftarrow CalcF()$

$Move(F)$

 iteration $\leftarrow$ iteration + 1

end while

Details of each procedure follow. *Initialize* is a procedure that aims to randomly generate m points from the feasible region. Each coordinate of a point x_k^i ($k = 1, \dots, n$) is assumed to be uniformly distributed between the corresponding upper and lower bounds, i.e., $x_k^i = l_k + \lambda (u_k - l_k)$ where $\lambda \sim U(0, 1)$. After computing the objective function value for all the points in the population, the procedure identifies the best point, x^{best} , which is the point with the best function value.

The *Local* procedure performs a local refinement and can be applied to one point or to all points in the population. The local search presented in [2] is a random line search algorithm that is applied coordinate by coordinate only to the best point in the population. First, the procedure computes the maximum feasible step length, $Length = \delta (max_k (u_k - l_k))$, based on δ . This quantity is used to guarantee that the local search generates feasible points. Second, the best point is assigned to a temporary point y to store the initial information. Next, for each coordinate k , a random number λ between 0 and 1 is selected as a step length and the point y_k is moved along that direction, $y_k = y_k + \lambda Length$. If an improvement is observed, within LSIt iterations, the best point is replaced by y and the search along that coordinate ends.

The *CalcF* procedure aims to compute the total force exerted on a point via other points. First a charge-like value, q^i , that determines the power of attraction or repulsion for the point x^i , is assigned. The charge of the point is calculated according to the relative efficiency of the objective function values, i.e.,

$$q^i = \exp \left(-n \frac{(f(x^i) - f_{best})}{\sum_{j=1}^m (f(x^j) - f_{best})} \right), \quad i = 1, \dots, m \quad (12.2)$$

Hence, the points that have better objective function values possess higher charges. The total force vector F^i exerted on each point is calculated by adding the individual component forces, F_j^i , between any pair of points x^i and x^j ,

$$F^i = \sum_{j \neq i}^m F_j^i = \begin{cases} (x^j - x^i) q^i q^j / (\|x^j - x^i\|^3), & \text{if } f(x^j) < f(x^i) \\ (x^i - x^j) q^i q^j / (\|x^j - x^i\|^3), & \text{if } f(x^j) \geq f(x^i) \end{cases} \quad (12.3)$$

$i = 1, \dots, m$.

Finally, the *Move* procedure uses the total force vector F^i , to move the point x^i in the direction of the force by a random step length,

$$x_k^i = x_k^i + \lambda \frac{F^i}{\|F^i\|} (RNG), i = 1, 2, \dots, m \text{ and } i \neq best \quad (12.4)$$

where RNG is a vector that contains the allowed range of movement towards the lower bound l_k , or the upper bound u_k , for each coordinate k . The random step length λ is assumed to be uniformly distributed between 0 and 1. Note that feasibility is maintained by using the normalized force exerted on each point. The best point, x^{best} , is not moved and is carried out to the subsequent iteration.

12.3 Hooke and Jeeves pattern search method

In this section, we describe our modification to the original EM algorithm. In this algorithm, the *Local* procedure is based on a random line search method [2]. Here a new *Local* procedure based on the Hooke and Jeeves (HJ) pattern search algorithm is proposed. This is a derivative-free method that searches in the neighbourhood of a point x^i for a better approximation via exploratory and pattern moves [4,6]. To reduce the number of function evaluations, the HJ pattern search algorithm is applied to the current best point only. This algorithm is a variant of the coordinate search, in the sense that incorporates a pattern move to accelerate the progress of the algorithm, by exploiting information obtained from the search in previous successful iterations. The exploratory move carries out a coordinate search (a search along the coordinate axes) about the best point, with a step length δ . If a new trial point, y , with a better function value than x^{best} is encountered, the iteration is successful. Otherwise, the iteration is unsuccessful and δ should be reduced. If the previous iteration was successful, the vector $y - x^{best}$ defines a promising direction and a pattern move is then implemented, meaning that the exploratory move is carried out about the trial

point $y + (y - x^{best})$, rather than about the current point y . Then, if the coordinate search is successful, the returned point is accepted as the new point; otherwise, the pattern move is rejected and the method reduces to coordinate search about y . Please see [4] for details. To ensure feasibility in the HJ pattern search algorithm an exact penalty strategy is used. This technique considers solving

$$\min F(x) \equiv \begin{cases} f(x) & \text{if } x \in \Omega \\ \infty & \text{otherwise} \end{cases} \quad (12.5)$$

rather than problem (12.1). This means that any infeasible trial point is rejected, since the objective function value is ∞ .

12.4 Numerical results

Computational tests were performed on a PC with a 3GHz Pentium IV microprocessor and 1Gb of memory. We compare the original EM algorithm, as described in Sect. 12.2, with the herein proposed EM algorithm modified with the HJ *Local* procedure, described in Sect. 12.3. We use a collection of 18 test functions [2,3,5] (see Table 12.1).

Table 12.1. Test functions and the corresponding parameters used by EM

Test function	n	Box constraints	f_{global}	m	MaxIt	LSIt	δ
Shekel5	4	$[0,10]^4$	-10.153200	40	150	10	1.00E-03
Shekel7	4	$[0,10]^4$	-10.402941	40	150	10	1.00E-03
Shekel10	4	$[0,10]^4$	-10.536410	40	150	10	1.00E-03
Hartman3	3	$[0,1]^3$	-3.862782	30	75	10	1.00E-03
Hartman6	6	$[0,1]^6$	-3.322368	30	75	10	1.00E-03
Goldstein-Price	2	$[-2,2]^2$	3.000000	20	50	10	1.00E-03
Branin	2	$[-5,10] \times [0,15]$	0.397887	20	50	10	1.00E-03
Six-hump Camel	2	$[-3,3] \times [-2,2]$	-1.031628	20	50	10	1.00E-03
Shubert	2	$[-10,10]^2$	-186.730909	20	50	10	1.00E-03
Griewank	2	$[-100,100]^2$	0.000000	30	100	20	1.00E-03
Himmelblau	2	$[-6,6]^2$	0.000000	10	50	5	1.00E-03
Sine envelope	2	$[-0.5,0.5]^2$	0.000000	20	75	10	5.00E-04
Bohachevsky	2	$[-10,10]^2$	0.000000	20	75	20	1.00E-03
Easom	2	$[-10,10]^2$	-1.000000	20	50	10	1.00E-03
Hump	2	$[-5,5]^2$	0.000000	20	50	10	1.00E-03
Spherical	2	$[-100,100]^2$	0.000000	30	75	20	1.00E-03
Three-hump	2	$[-5,5]^2$	0.000000	20	50	10	1.00E-03
Zakharov4	4	$[-5,10]^4$	0.000000	30	75	20	1.00E-03

The first four columns of the table refer to the name of the function, the dimension of the problem, n , the default box constraints, and the known global optimum, f_{global} . The last four columns list the parameters used by EM for each function.

The results obtained by the original EM algorithm are shown in Table 12.2 and the results of the EM algorithm modified with the HJ *Local* procedure are presented in Table 12.3. We use average results for comparison, over 25 runs. Table 12.2 and Table 12.3 report average number of function evaluations, Ev_{avg} , the average best function values, f_{avg} , and the best function value, $f_{best} = \min(f_{best}^i, i = 1, \dots, nruns)$, over $nruns=25$ runs. Values of the mean absolute error

$$MAE = \frac{|f_{avg} - f_{global}|}{n} \quad (12.6)$$

and the standard deviation

$$SD = \sqrt{\frac{\sum_{i=1}^{nruns} (f_{best}^i - f_{avg})^2}{nruns}} \quad (12.7)$$

are also listed for each problem.

Table 12.2. Results of original EM with *Local* procedure applied to the best point

Test function	Ev_{avg}	f_{avg}	f_{best}	MAE	SD
Shekel5*	1865	-10.152770	-10.153163	0.000108	0.000270
Shekel7	1480	-10.402471	-10.402884	0.000117	0.000486
Shekel10	1486	-10.535939	-10.536256	0.000118	0.000243
Hartman3	1260	-3.862487	-3.862722	0.000098	0.000111
Hartman6	1850	-3.322267	-3.322365	0.000017	0.000091
Goldstein Price	488	3.000141	3.000001	0.000071	0.000077
Branin	512	0.397906	0.397888	0.000009	0.000012
Six Hump Camel	291	-1.031599	-1.031628	0.000015	0.000026
Shubert	357	-186.723068	-186.730906	0.003920	0.005750
Griewank**	1582	0.000062	0.000008	0.000031	0.000049
Himmelblau	347	0.000047	0.000008	0.000023	0.000023
Sine envelope	518	0.000044	0.000002	0.000022	0.000027
Bohachevsky	793	0.000063	0.000001	0.000032	0.000041
Easom	489	-0.999966	-1.000000	0.000017	0.000029
Hump	287	0.000044	0.000001	0.000022	0.000027
Spherical	903	0.000033	0.000001	0.000017	0.000030
Three-hump	382	0.000041	0.000001	0.000021	0.000030
Zakharov4	1621	0.000068	0.000014	0.000017	0.000022

The termination criteria and the used parameters are the ones proposed in [2]. Thus, both algorithms stop when the number of iterations exceeds MaxIt unless the relative error in the best objective function value, with respect to f_{global} , is less than 0.01%. In the HJ algorithm, the factor used to reduce δ , whenever an unsuccessful iteration is found, is 0.1 and the minimum step length allowed was 1×10^{-8} . In the tables, * means that 4 (in Table 12.1) and 5 (in Table 12.2) runs of Shekel5 did not converge and ** means that 3 runs of Griewank did not converge.

The results obtained with the EM algorithm modified with the HJ *Local* procedure are better than the ones produced by the algorithm of Sect. 12.2, as far as the accuracy of the results is concerned. The proposed algorithm achieves in general the lowest numerical errors (MAE) and lowest standard deviations (SD) for 25 runs. However, Table 12.3 reveals in some cases larger number of function evaluations.

Table 12.3. Results of EM with HJ *Local* procedure applied to the best point

Test function	Ev_{avg}	f_{avg}	f_{best}	MAE	SD
Shekel5*	1880	-10.153169	-10.153186	0.000008	0.000023
Shekel7	1677	-10.402855	-10.402934	0.000022	0.000205
Shekel10	1679	-10.536320	-10.536405	0.000022	0.000177
Hartman3	1482	-3.862541	-3.862775	0.000080	0.000106
Hartman6	2378	-3.322308	-3.322361	0.000010	0.000075
Goldstein Price	405	3.000036	3.000001	0.000018	0.000054
Branin	372	0.397900	0.397888	0.000006	0.000012
Six Hump Camel	261	-1.031588	-1.031628	0.000020	0.000025
Shubert	641	-186.714293	-186.730827	0.008308	0.067471
Griewank**	2095	0.000018	0.000000	0.000009	0.000017
Himmelblau	294	0.000018	0.000000	0.000009	0.000021
Sine envelope	423	0.000025	0.000000	0.000012	0.000030
Bohachevsky	1892	0.000018	0.000000	0.000009	0.000011
Easom	476	-0.999965	-1.000000	0.000018	0.000033
Hump	315	0.000023	0.000000	0.000012	0.000032
Spherical	514	0.000031	0.000000	0.000015	0.000032
Three-hump	337	0.000027	0.000001	0.000013	0.000021
Zakharov4	1430	0.000013	0.000001	0.000003	0.000027

12.5 Conclusions

We have studied the Electromagnetism-like algorithm and implemented a new *Local* procedure based on a pattern search method. This new algorithm was applied to different test problems from the literature and com-

pared with the original EM. The preliminary results seem promising. Future developments will focus on extending the numerical experiments to other test functions with larger dimensions and many local minimizers in the feasible region. The purpose here is to analyze the pattern search ability to drive the best point towards the global minimizer instead of to a non-global one.

References

1. Birbil SI (2002) Stochastic global optimization techniques. PhD Thesis, North Carolina State University
2. Birbil SI, Fang S (2003) An electromagnetism-like mechanism for global optimization. *J. of Global Optimization* 25(3):263-282
3. Dixon LCW, Szegö GP (1978) The global optimization problem: an introduction. In: Dixon LCW, Szegö GP (eds) *Towards Global Optimisation 2*. North-Holland, Amsterdam, pp 1-15
4. Hooke R, Jeeves TA (1961) Direct search solution of numerical and statistical problems. *J. Assoc. Comput. Mach.* 8:212-229
5. Huyer W, Neumaier A (1999) Global optimization by multilevel coordinate search. *J. Global Optimization* 14:331-355
6. Lewis RM, Torczon V (1999) Pattern search algorithms for bound constrained minimization. *SIAM J. Optim.* 9(4):1082-1099

Chapter 13

Latency hiding and adaptability for parallel iterative algorithms

Javier Roca, José Antonio Álvarez, Jose Jesús Fernández

Departamento de Arquitectura de Computadores y Electrónica.
Universidad de Almería {jroca|jaberme|jose}@ace.ual.es

Abstract. Switching from the MPI process model to a threaded model in the parallel environment via user level threads takes advantage of the existing concurrence in applications. Major aims for this work are to face the computational demands of scientific applications by exploiting concurrency, minimizing latency due to message passing, and to present and evaluate strategies to implement adaptability when imbalance occurs.

Keywords. User level threads, Parallel iterative reconstruction algorithms, Dynamic Load Balancing

13.1 Introduction

In contrast to the legacy and structured programming model used in standard MPI based applications, multithreaded programming [1] provides a way to divide a program into different pieces that can be executed concurrently. Moreover using user level threads [2] in places where concurrence can be exploited would allow us to achieve latency hiding, better scalability, skills to avoid overhead on processors due to faster context switching and abilities to migrate threads for load balancing purposes. On the other hand, data locality is an important issue that ties up a large amount of scientific applications. Image processing such as filtering or 3D tomographic

reconstruction algorithms [3], are examples. In the design of a parallel strategy for this kind of applications, data distributions have to be devised so that the locality is preserved as much as possible. Otherwise, the application will exhibit poor performance due to excessive communications. Data distributions that preserve locality are specially important in systems where the workload is dynamically reassigned.

Our application, *BICAV* [4], addresses the tomographic reconstruction problem using iterative methods. These methods are far superior to the standard technique, weighted backprojection, in terms of reconstruction quality [4]. However, they are much more computationally demanding.

In this work, we present and evaluate a parallel implementation of iterative reconstruction methods using a multithreaded approach with AMPI [5]. This framework embeds MPI processes into user level threads, a more efficient technique in aspects like context switching, migration of tasks to other processors. The aim is to face the computational demands of these methods by exploiting concurrency and minimizing latency due to message passing. Evaluating the efficiency achieved when applying adaptability techniques, in scenarios where load imbalance -either intrinsic or extrinsic- arises, is another major objective. For this purpose, a particular implementation of the 3D Jacobi algorithm [6], a classical iterative method for solving partial differential equations that exhibits strong locality properties alike -when data is distributed along only one dimension- to those presented by *BICAV*. Jacobi is, therefore, used to test the load balancing strategy implemented, that seizes data locality properties inherent to aforementioned applications. Experiments performed for Jacobi show gains and key spots to work on when using load balancing strategy with our tomographic applications.

This paper is organized as follows. Section 13.2 analyzes the iterative reconstruction techniques and its data dependences. Section 13.3 describes the adaptability issues aroused when load balancing is applied. Section 13.4 analyzes how the threaded implementation behaves when compared with its MPI version, and also describes and compares the load balancing strategies. Finally, in last section, conclusions are exposed.

13.2 Parallel iterative reconstruction methods

Series expansion reconstruction methods assume that the 3D object, or function f , can be approximated by a linear combination of a finite set of known and fixed basis functions, with density x_j . The aim is to estimate the unknowns, x_j . These methods are based on an image formation model

where the measurements depend linearly on the object in such a way that $y_i = \sum_{j=1}^J l_{i,j} \cdot x_j$, where y_i denotes the i^{th} measurement of f and $l_{i,j}$ the value of the i^{th} projection of the j^{th} basis function. Under those assumptions, the image reconstruction problem can be modeled as the inverse problem of estimating the x_j 's from the y_i 's by solving the system of linear equations aforementioned.

Assuming that the whole set of equations in the linear system may be subdivided into B blocks, a generalized version of component averaging methods, *BICAV*, can be described. The processing of all the equations in one of the blocks produces a new estimate. All blocks are processed in one iteration of the algorithm. This technique produces iterates which converge to a weighted least squares solution of the system.

A volume can be considered made up of 2D slices. When voxel basis functions are used, the slices are independent. However, the use of the spherically symmetric volume elements (blobs) [7], makes slices interdependent because of blob's overlapping nature.

We have implemented the parallel iterative reconstruction method adapting legacy MPI code into a multithreaded environment by means of Adaptive MPI (AMPI). The block iterative version of the component averaging methods, *BICAV*, has been parallelized following the Single Program Multiple Data (SPMD) approach [4]. The volume is decomposed into slabs of slices that will be distributed across the nodes (for MPI) or threads (for AMPI). The threads can, thereby, process their own data subdomain, however, the interdependence among neighbor slices due to the blob extension makes necessary the inclusion of redundant slices into the slabs. In addition, there must be a proper exchange of information between neighbor nodes to compute the forward-projection or the error back-projection for a given slab and to keep redundant slices updated. These communication points constitute synchronization points in every pass of the algorithm in which the nodes must wait for the neighbors. The amount of communications is proportional to the number of blocks and iterations. Reconstruction yields better results as the number of blocks is increased.

In any parallelization project where communication between nodes is involved, latency hiding becomes an issue. That term stands for overlapping communication and computation so as to keep the processor busy while waiting for the communications to be completed. Our improvement proposal achieves latency hiding via user level threads.

13.3 Adaptability

In general, load balancing strategies are based on the distinction of heavily versus lightly loaded processors at the moment in which load balancing decision is taken by the system. The main issue consists then of migrating tasks among processors so that the nodes in the system get an even workload. Nevertheless, in general, such strategies do not take into consideration data locality, and they may assign pieces of work that share data to different processors. Consequently, an unnecessary increment of communications flow would appear.

The strategy for locality-preserving load balancing (RakeLB) was implemented as a centralized load balancer. For RakeLB, two processors are neighbours (adjacent) if both have at least a thread with data in common. RakeLB has been implemented taking advantage of the migration facilities, also, provided by AMPI. Then the migration of workload among nodes is carried out in terms of threads. Employing user level threads make load balancing issues easier in SPMD applications. Threads owns an universal ID, the rank, this rank establishes an order relationship between them. When computation is ignited, each thread picks up a chunk of data to work on, therefore, threads with consecutive ranks will need, at some point in time, communicate one each other. Therefore, thread migration decisions should conserve as much as possible the established thread order. A FIFO class structure keeps threads ordered during migrations. When threads migration is advised, those with the minimal data locality restrictions in the node (the ones occupying the end of the stack) will be moved. RakeLB behaviour has been contrasted with a, standard centralized, dynamic load balancing strategy, like Greedy [8].

13.4 Results

The platform used to evaluate our algorithms was our research group cluster, VERMEER, which owns as a front-end a dual Pentium IV XEON 3.2 Ghz, 512KB Cache node with Gentoo Linux, Kernel 2.4.32 SMP and two Pentium IV XEON 3.06 Ghz, 512KB L2 Cache per computing node with 2GB of ECC DDR SDRAM, connected via 1Gb Ethernet.

This work pursuits the study of the gain obtained by the multithreaded implementation of *BICAV*, which overlaps computation/communication. Comparison to its MPI version, where latency hiding is achieved using advanced programming techniques, to interleave computations and communication, was also aimed. Thereby, scaling tests were carried out varying

the number of threads/processor and the number of processors, for both versions AMPI and MPI. As a concluding phase, the load balancing strategy developed (RakeLB), is evaluated and compared to Greedy, a standard strategy. The initial set of experiments were intended to quantify the influence of the number of blocks over the global performance. As the number of blocks increases, the convergence of the algorithm is faster, but the amount of communications also increases. This test was performed on the Vermeer cluster using up to 32 physical processors.

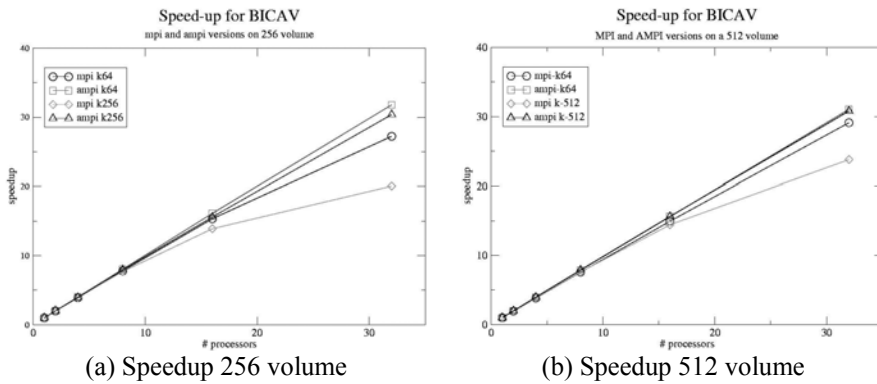


Fig. 13.1. Speedup in Vermeer cluster for volumes of dimensions 256 and 512

Figure 13.1 shows the speedup of the MPI version and the AMPI version with 128 threads, for two different volumes (256 and 512), using several numbers of blocks K . It can be observed that below a threshold of $K=64$ both versions seem to behave similarly, showing slight improvement in AMPI. But above that threshold, and as K is increased, AMPI behaves better than MPI especially for more than 16 processors. AMPI is getting benefits from the hidden concurrency by means of an intelligent overlapping of running thread communication and sleeping threads' computation still to be done. Speedup curves in Figure 13.1 clearly show that the application implemented with MPI loses the linear speedup when using more than eight processors. However, AMPI keeps speedup almost linear up to 32 processors. Therefore, thread switching succeeds in maintaining an optimal speedup.

In the subsequent sets of experiments, the 3D version of the Jacobi algorithm was used because it exhibits stronger data dependence and, hence, emphasizes the influence of data locality preservation on the performance. As the bigger number of threads, are created, the higher accuracy we obtain, also computational load increases. Tests carried out were characterized by Jacobi performing its work along ten iterations and the load balancer strategy being invoked three times. In order to simulate imbalanced

scenarios, processors were loaded with random workload (background load). At the initial step, threads from our application were evenly distributed among nodes.

A formal prove of imbalance was carried out by employing as a metric the standard deviation of the whole system load, normalized to the average load, σ . The initial imbalance value reflected by σ was over 0.5. After load balancing, both strategies achieved a similar σ value (0.051 for GreedyLB and 0.045 for RakeLB), but there's a key point not stated yet, *data dependency*.

After load balancing, a complete homogeneous load distribution is achieved (σ almost zero). Although load balancing strategies get an almost alike load distribution, it is key to note that these distributions are completely different, in terms of data locality, aspect that turns to be an issue for performance when for applications data locality is a significant characteristic.

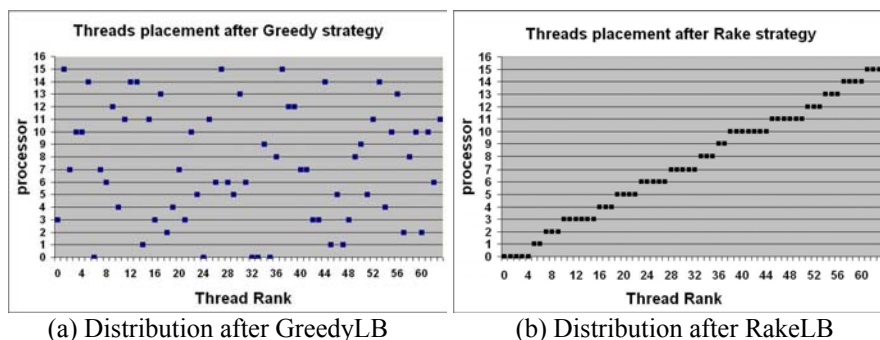


Fig. 13.2. Thread placement, before and after the first balancer call

Figure 13.2 intends to show that although each load balancer did a reasonable load distribution to respond to the imbalanced situation, for this application, that was not enough. RakeLB algorithm preserved data locality, see Figure 13.2(b), it can be seen how the neighborhood relationship between threads was conserved. GreedyLB strategy Figure 13.2(a) balanced load without paying attention to locality issues, and not preserve the original ordering among threads when they migrate to other nodes. As stated before, the version of Jacobi used is characterized by a proportional growth of computation as the number of threads is increased. As seen in Figure 13.3, scaling for the application being balanced with RakeLB experiments a better behaviour than its counterpart strategy. In every test, keeping the maximum number of neighboring threads in the same processor as RakeLB does, leads to a performance improvement in the application. This performance improvement begins to be much more significant for Jacobi

when balanced with the RakeLB strategy in contrast with the cases where Jacobi was balanced with Greedy strategy.

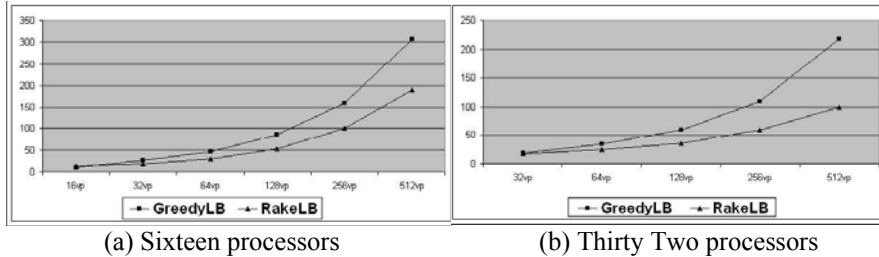


Fig. 13.3. Performance of Jacobi when data locality based strategies are used

13.5 Discussion and conclusions

Through experiments, using AMPI, we illustrated how one of our scientific iterative applications was directly benefited as the cpu and walltime gap per iteration was almost zero, due to an optimal concurrence exploitation for the threaded version, in contrast to the MPI version which did not respond so well.

On one hand the performance of this technique, was analyzed, varying the number of physical processors and the population of threads. It could, therefore, be stated that the threaded version of *BICAV* scaled significantly better than the MPI version. From these experiments we can conclude that user level threads are an attractive approach for programming parallel scientific applications, which can be benefited from having multiple flows of control. On the other hand, the dynamic load balancing strategy which preserves data locality, was proved to be efficient for applications like *BICAV*. Currently, we are applying and testing RakeLB in our parallel reconstruction software.

Acknowledgement. Work partially supported through grants MEC-TIN 2005-00447 and JA-P06-TIC-01426

References

1. Pancake CM (1993) Multithreaded languages for scientific and technical computing. In: Proceedings of the IEEE 81(2)

2. Price GW, Lowenthal DK (2003) A comparative analysis of fine-grain threads packages. *Journal of Parallel and Distributed Computing*. College Station, Texas, 63:1050-1063
3. Kak AC, Slaney M (2001) *Principles of computerized tomographic imaging*. SIAM Society for Industrial and Applied Mathematics
4. Fernández J, Lawrence AF, Roca J, García I, Ellisman MH, Carazo JM (2002) High performance electron tomography of complex biological specimens. *Journal of Structural Biology* 138:6-20
5. Huang Ch, Zheng G, Kumar S, Kale LV (2006) Performance evaluation of Adaptive MPI. In: *Proceedings of ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*
6. Press W, Teukolsky S, Vetterling W, Flannery B (1992) *Numerical recipes in C. The art of scientific computing*. Cambridge University Press
7. Matej S, Lewitt R, Herman G (1996) Practical considerations for 3-D image reconstruction using spherically symmetric volume elements. *IEEE Trans. Med. Imag.* 15:68-78
8. Aggarwal G, Motwani R, Zhu (2006) The load rebalancing problem. *Journal Algorithms* 60:42-59

Chapter 14

Clustering the linearly inseparable clusters

Hamid Reza Shahdoosti, Omid Khayat

Department of Biomedical Engineering, Amirkabir University, IRAN

Abstract. In this paper a new method for clustering is introduced that works fine for linearly inseparable clusters. This algorithm uses the of patterns' concentration specification. We consider each pattern as a heat source to determine concentration by heat effect. Because of high concentration in clusters, heat in the clusters is higher than out of them, and this fact is used to determine and bound the clusters. CLIC algorithm can be implemented for any linearly inseparable cluster. The classification rate of this algorithm is higher than previously known methods (k-means, fuzzy c-means, etc). Finally, we cluster real iris data by using CLIC algorithm.

Keywords. Clustering, classification, heat function, hyper plane, iris data

14.1 Introduction

There are several methods of pattern clustering and classification. Clustering and classification are among the most important topics in artificial intelligence, such as signal processing, image processing, pattern recognition. Several popular methods are known for clustering and classification such as k-means, nearest neighbor and fuzzy c-means that are based on Euclidean distance. CLIC algorithm presents a new method for clustering and classification that is based on defining the heat function. We have a large number of patterns and want to cluster them into a desired number of clusters. We consider each pattern as a heat source in the N-dimensional

feature space. Each pattern affects the whole N-dimensional feature space; this which causes the points in the N-dimensional feature space to be affected by the all patterns. CLIC algorithm finds the hyper-planes in the N-dimensional feature space that separate the clusters, based on the heat intensity. If several linearly inseparable clusters exist, the algorithm enables us to separate them two by two. The classification rate of this algorithm is higher than the classification rate of previously known methods, such as k-means, nearest neighbor, fuzzy c-means. In the following sections the algorithm is explained in more details.

14.2 Description of the algorithm

We have two clusters in the N-dimensional feature space, which are not linearly separable. Consider the following clusters:

$$X = \{x_1, x_2, \dots, x_n\} \quad Y = \{y_1, y_2, \dots, y_n\} \quad (14.1)$$

$$x_i, y_i = (d_1, d_2, \dots, d_N)_{(i=1,2,\dots,n)}$$

where X and Y are collections of patterns for first and second clusters respectively, and d_j ($j=1\dots N$) is the j th feature of patterns. Since the close patterns cause concentration, it is necessary to determine high and low concentration of patterns in the space. We consider each pattern as a heat source to determine concentration by heat effect. Patterns (heat sources) affect the whole space. The interaction of near patterns (in a cluster) is considerable. On the other hand, it is negligible when the patterns are far from each other. This means the effect decreases when the distance increases. By considering this effect as the heat, we define heat function as below:

$$H(X_0|X_j) = F(\text{DISTANCE BETWEEN } X_0 \text{ \& } X_j), (j=1,2,\dots,k) \quad (14.2)$$

where $H(X_0|X_j)$ is the heat of arbitrary point X_0 affected by pattern X_j , which depends on the distance between point X_0 and pattern X_j . F is a function of distance that satisfies 3 conditions:

1. $F(0) < \infty$
2. $F(\infty) \rightarrow 0$
3. if $d < d' \rightarrow F(d) > F(d')$

The first condition points to the heat limitation in patterns, the second condition expresses that the heat is negligible for high distances, and the third shows that the function is decreasing by distance.

By increasing the distance from each pattern, heat effectiveness of pattern decreases by a decreasing function (for example exponential function). It is obvious that intensity of heat could be explanatory of high or low concentration of patterns. The heat can be calculated for each arbitrary point X_0 affected by K patterns as below:

$$H(X_0) = \sum_{j=1}^k H(X_0 | X_j) \quad (14.3)$$

Here, $H(X_0 | X_j)$ is the heat of point X_0 affected by pattern X_j , and $H(X_0)$ is the total heat of arbitrary point X_0 (affected by all patterns).

First, it is needed to define a decreasing function and use it as F . Depending on conditions, the decreasing function F can be determined. We can determine the decreasing function F , by considering the effect of each pattern on patterns in other clusters should be negligible. By having the necessary decrease in each direction we can determine the type of function (exponential or any other nonlinear decreasing function) and the coefficients of the function. For example, when clusters are close, we consider F high decreasing function to have low heat between clusters (versus in the clusters).

Now we want to find the HYPER-PLANE in the N -dimensional space, which separates the clusters. A hyper-plane in N -dimensional space is defined as below:

$$a_1 d_1 + a_2 d_2 + \dots + a_N d_N = a_0 \quad (14.4)$$

In which d_i are coordination axis, a_i are coefficients, and a_0 is equation's constant. By varying the coefficients, the equation can embrace all hyper-planes in the space. Since the coefficients have no limitation and they are unlimited we can replace them with $\tan \theta_i$, in which $-\pi/2 < \theta_i < \pi/2$:

$$a_i \approx \tan(\theta_i), \quad -\infty < a_i < \infty \rightarrow -\pi/2 < \theta_i < \pi/2 \quad (14.5)$$

By replacing a_i with $\tan \theta_i$ we have:

$$\tan \theta_1 . d_1 + \tan \theta_2 . d_2 + \dots + \tan \theta_N . d_N = \sum_{i=1}^N \tan \theta_i . d_i = a_0, \quad -\pi/2 < \theta_i < \pi/2 \quad (14.6)$$

By discriminating the θ_i , the hyper-plane rotates step by step, and the step size affects the accuracy and complexity of problem.

Considering the fact that the heat of each point is affected by all patterns, we can calculate the heat of hyper-plane as below:

$$H(\text{hyper plane}) = \int_{d_{1\min}}^{d_{1\max}} \int_{d_{2\min}}^{d_{2\max}} \dots \int_{d_{N\min}}^{d_{N\max}} \left(\sum_{j=1}^n H(ds|X_j) \right) ds \quad (14.7)$$

where $H(\text{hyper plane})$ is the heat of hyper-plane, ds is the unit of surface on the hyper-plane that its heat depends on distance between center of unit with all patterns. $d_{i\min}$ and $d_{i\max}$ determine the limitations of integral that are being determined based on the intensity of heat decrease around the clusters. The goal is finding the best hyper-plane which can separate the clusters. Each state of hyper plane has certain $\theta_1, \theta_2, \theta_3, \theta_4$. By varying the angles with an identified step ($\Delta\theta_i$), the state of hyper plane varies. At each state of hyper-plane (with an identified angles), we move the hyper-plane in the direction of hyper plane's normal vector to cross the clusters. Then we plot the diagram of heat of hyper plane (at each move) versus the number of patterns distanced by hyper-plane. Some diagrams of hyper-plane's heat (relative to the number of distanced patterns), have two main maxima (main maximum occurs when hyper plane travels through each cluster center) and a main minimum (main minimum occurs when hyper plane traverses between clusters) between them. Of course, the diagram may have one main maximum (has no main minimum) when hyper-plane traverses two cluster's center together (almost). Now we consider all diagrams have a main minimum, and find the diagram that has lowest main minimum. Assume that the hyper plane with lowest minimum is as below:

$$\tan \theta_1 . d_1 + \tan \theta_2 . d_2 + \dots + \tan \theta_N . d_N = a_0 \quad (14.8)$$

Whereby $\theta_{i(i=1,2,\dots,N)}$ and a_0 are certain constants. The hyper- plane with the lowest heat (that stands between centers) is shown in equation above. The hyper-plane separates two clusters. If we assume the number of pattern assigned to first cluster is n'_1 and the number of pattern assigned to second cluster is n'_2 , 3 cases may occur:

- First case: n_1, n_2 are not identified, so consider the hyper-plane mentioned as a cluster separator, that results two clusters are specific.

- Second case: n_1, n_2 are identified, and $n'_1 = n_1$, $n'_2 = n_2$. Similar to first state we consider the mentioned hyper plane as a cluster separator, that results two clusters are specific.
- Third case: n_1, n_2 are identified, $n'_1 \neq n_1$ and $n'_2 \neq n_2$. Assuming $n'_1 < n_1$, we need to choose the difference number of patterns from second cluster and assign them to first cluster. We consider heat distribution of the mentioned hyper-plane (the hyper plane with lowest heat), assume the point with highest heat on the hyper plane is $S = (d_1, d_2, \dots, d_N)$ and the necessary number of patterns is $n_1 - n'_1 = n''$, now find the n'' nearest patterns to point S and assign them to first cluster. So the number of assigned patterns to first cluster (by algorithm) becomes equal to the real number of patterns (also for second cluster).

14.3 Iris data

To show that our algorithm also work fine in real data set, we use the famous Iris plant data set. The data set contains 3 classes of 50 instances each, where each class refers to a type of Iris plant: Setosa, Versicolor, and Virginica. The attributes consist of the Sepal Length, Sepal Width, Petal Length and Petal Width. One class (red patterns) is linearly separable from the other two; the latter are not linearly separable from each other (green and blue patterns). Iris data is shown in Figure 14.1. We want to separate them by algorithm.

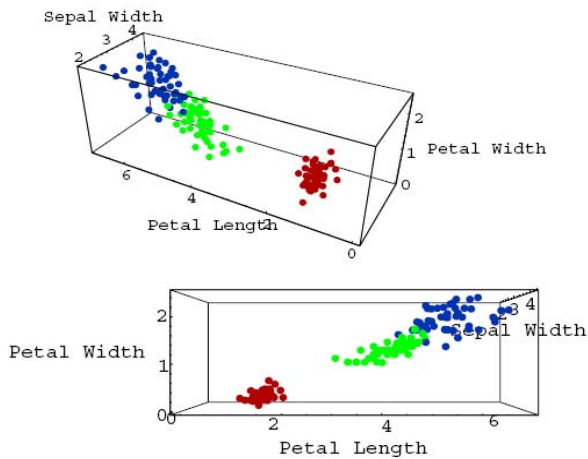


Fig. 14.1. Iris data

First we need to define a decreasing function and use it as F . An appropriate function is exponential function. So we can calculate the heat of arbitrary point X_0 affected by pattern X_j as below:

$$H(X_0|X_j) = \text{Exp}\left(-\sum_{k=1}^N \mu_k (d_{kx_0} - d_{kx_j})^2\right) \quad (14.9)$$

$$x_0 = (d_{1x_0}, d_{2x_0}, \dots, d_{Nx_0})$$

$$x_j = (d_{1x_j}, d_{2x_j}, \dots, d_{Nx_j})$$

That $\mu_{i(i=1,2,\dots,N)}$ is coefficient of distance at direction d_i , and X_0 is an arbitrary point and X_j is a pattern.

Now consider two linearly inseparable clusters that are shown in Figure 14.1 with blue and green patterns. Noticing the average distance between patterns in the same clusters and distance between clusters, we put $\mu_1 = \mu_2 = \mu_3 = \mu_4 = 5$ (the coefficient $\mu = 5$ is an appropriate coefficient). Now we consider the hyper plane in 4-dimensional coordination.

By varying the angle $\theta_{i(i=1,2,\dots,N)}$ (with an identified gradient $\Delta\theta_i$) with certain $\mu_{i(i=1,2,\dots,N)}$ the hyper plane with minimum heat belongs to diagram Figure 14.2.

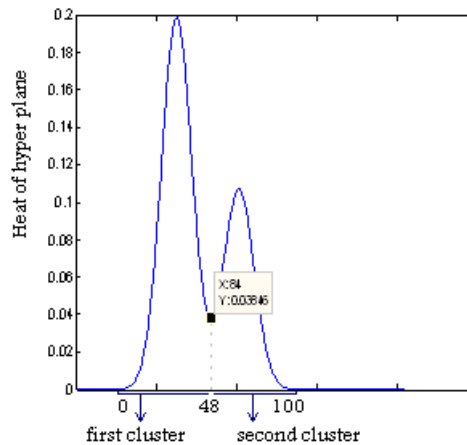


Fig. 14.2. The diagram of heat of hyper plane versus number of patterns distanced $\theta_1 = -30^\circ, \theta_2 = 25^\circ, \theta_3 = 45^\circ, \theta_4 = 45^\circ$, $\mu_1 = \mu_2 = \mu_3 = \mu_4 = 5$, $a_0 = 4.25$ $\Delta\theta = 5^\circ$

The number of assigned pattern to first cluster by the hyper plane with minimum heat is 48 patterns in which 47 patterns belong to the first clus-

ter. We find two nearest patterns (the real number of first cluster is 50) to the point with maximum heat on the hyper plane. Two found patterns belong to the first cluster, so we have:

$$\text{classification rate} = \frac{\text{true assigned patterns}}{\text{Total of patterns}} = 98.66\% \quad (14.10)$$

Table 14.1. The value of classification rate relative to coefficients $\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu$ and gradient of angle ($\Delta\theta_1 = \Delta\theta_2 = \Delta\theta_3 = \Delta\theta_4 = \Delta\theta$)

	$\Delta\theta = 30^\circ$	$\Delta\theta = 15^\circ$	$\Delta\theta = 10^\circ$	$\Delta\theta = 5^\circ$	$\Delta\theta = 2^\circ$	$\Delta\theta = 1^\circ$
$\mu = 0.3$	73.33%	80%	85.33%	89.33%	90.66%	90.66%
$\mu = 1$	74.66%	90.66%	90.66%	93.33%	96%	96%
$\mu = 5$	82.66%	93.33%	97.33%	98.66%	98.66%	98.66%
$\mu = 20$	80%	90.66%	93.33%	96%	97.33%	97.33%

14.4 Conclusion

Noticing the result of clustering based on CLIC algorithm, we find CLIC algorithm as a strong and useful algorithm. The methods only based on Euclidean distance like k-means, nearest neighbor, fuzzy c-means are uninformed methods which need high memory capacity. CLIC algorithm is based on defining the heat function, it can use program knowledge. The algorithm introduced (CLIC) is an informed algorithm. That result a high classification rate more than the classification rate of methods only based on Euclidean distance (fuzzy c-means, nearest neighbor, etc).

References

1. Rose Keneth, (1998) Deterministic annealing for clustering, compression, classification, regression, and related optimization problems.
2. Osipova Nataly (2004). Classification and clustering methods development and implementation for unstructured documents collections. Department of Programming Technology, Faculty of Applied Mathematics and Control Processes, St. Petersburg State University
3. Wallace Manolis, Kollias Stefanos (2004) Robust, generalized, quick and efficient agglomerative clustering.
4. Wolkenhauer Olaf Fuzzy classification, the iris-and-admission data sets.

Chapter 15

A top-down minimization algorithm for non-deterministic systems

Adrian Zafiu,¹ Stefan Holban²

¹Department of Electronics, Communications and Computers, University of Pitesti, 110040 Pitesti Str. Targul din Vale, nr.1, Arges, Romania, adrian_zafiu@yahoo.com, adrian.zafiu@upit.ro

²Department of Computer Science, University “Politehnica” of Timisoara, 300006 Timisoara Pta Victoriei No. 2, Timis, Romania, stefan@aspc.cs.upt.ro

Abstract. Nowadays, the decisional diagrams principle is general adopted, as the base technique, for decisional systems analysis. The proposed technique is based on different principles and strategies. This technique open the way for more native interpretations of logic comprised by a decisional system, free by constriction of hardware implementation possibilities.

Keywords. Multi-valued combinational minimization, generative method, processing redundancy, evaluation, implicants, network and parallel minimization, deterministic and nondeterministic networks

15.1 Introduction

There are no methods that can process the minimization into a polynomial time. There are many algorithms or methods approaching the problematic of minimization of multi-valued or binary decisional systems, like graphics systems (Veitch-Karnaugh diagrams), Quine McCluskey [1] method, Espresso [2],[3] and SIS[4]. The SIS software also minimizes multi-valued

specifications, but in a binary way, coding the multi-valued variables. This method is a pseudo multi-valued one. The MVSIS [5],[6] application minimizes multi-valued systems in a natural manner. BOOM [7] and BOOM-II [8] tools perform a heuristic implementation with outstanding results regarding the execution time, but applicable only for binary specifications. The Discrimination method [9] is the last known exact method for the multi-valued minimization.

Presented algorithm use a different principle named discrimination [9]. According to this principle we search the set of difference between groups of input combinations with the same output.

15.2 Method presentation

The minimization algorithm presented here requires an input table with disjointed rows. The selected minimal set of implicants is processed next to eliminate redundant if there are. The method consists from three main steps: pre-processing, generating the complete set of implicant vectors and selecting a non-redundant solution.

The preprocessing: The given specification must be preprocessed to obtain an equivalent table appropriate for the next step. In this stage, there is eliminated implicit non-determinism, composed vectors are translated into primes vectors and, if is possible, a superficial compression is performed.

Disjoint rows: Firstly, any pair of rows, which have implicit non-determinism, is decomposed. The results of decomposition transform the implicit non-determinism into an explicit one.

Let considering next function with default output $f : Z_3 \times Z_3 \times Z_3 \times Z_3 \rightarrow Z_2 \times Z_3$:

Table 15.1. $f : Z_3 \times Z_3 \times Z_3 \times Z_3 \rightarrow Z_2 \times Z_3$

#	i_1	i_2	i_3	i_4	/	o_1	o_2
1	1	-	2	-	/	1	0
2	-	2	-	0	/	1	1
3	2	-	2	2	/	-	2
4					/	0	-

Rows 1 and 2 have an implicit non-determinism because the input combination (1220), common for both rows, has output values 0 and 1 for second output variable. The decomposition is applied for each pair of rows, initial or resulted rows, with implicit non-determinism. The result of

decomposition is a set of rows with explicit non-determinism (row 1 of the result):

Table 15.2. Decomposition result for rows 1 and 2

#	i_1	i_2	i_3	i_4	o_1	o_2
1	1	2	2	0	1	0,1
2	1	0,1	2	1,2	1	0
3	1	0,1	2	0	1	0
4	1	2	2	1,2	1	0
5	0,2	2	0,1	0	1	1
6	0,2	2	2	0	1	1
7	1	2	0,1	0	1	1

Split composed rows into primaries rows:

Table 15.3. Prime input vectors

#	Row	#	Row	#	Row
1	1 2 2 0 / 1 0,1	6	1 0 2 0 / 1 0	11	2 2 0 0 / 1 1
2	1 0 2 1 / 1 0	7	1 1 2 0 / 1 0	12	0 2 1 0 / 1 1
3	1 1 2 1 / 1 0	8	1 2 2 1 / 1 0	13	2 2 1 0 / 1 1
4	1 0 2 2 / 1 0	9	1 2 2 2 / 1 0	14	0 2 2 0 / 1 1
5	1 1 2 2 / 1 0	10	0 2 0 0 / 1 1	15	2 2 2 0 / 1 1

Light compression: The last operation of the preprocessing consists in a light compression of rows. Row 0 2 – 0 / 1 1, for example, replaces rows 10, 12 and 14. This process avoid redundant computations without denaturize the results.

By considering all this operations, the specification presented in Table 15.2 becomes:

The implicants computation: This stage of processing generates a complete set of implicant vectors having a minimum number of literals (in two steps). The algorithm generates implicants with only one non-DC position and, if there remain uncovered rows, it continues to find implicants with two non-DC positions, and so on. By using this principle, is obvious that only necessary implicants with a minimum number of non-DC positions form the results.

The generator: The generator uses packages, which split the search space into disjunctive zones. The candidates comprised into a package have at least a common component that separates this candidate from the candidates comprised into other packages.

Let consider a function defined on the domain $Z_3 \times Z_3 \times Z_3$. The generator creates a candidate vector for each value of each input sub domain. First line of the following table contains the first package.

Table 15.4. The generator

Pack	1	2	3	4	5	6	7	8	9
1	0--	1--	2--	-0-	-1-	-2-	--0	--1	--2
1.1	00-	01-	02-	0-0	0-1	0-2			
1.2	10-	11-	12-	1-0	1-1	1-2			

The second package (1.1) is generated from the package 1 by preserving it's elements. First candidate is combined with candidates from 4 to 9 of first package. Items 2 and 3 cannot be combined with first candidate, because first position must have the same specified value.

The attribute computation: For each candidate, a set of attributes is computed, in order to characterize the candidate status. The set of attributes has a part commonly used for all output functions of each row and a part which is specific to each output function and to each output row.

The attribute $\langle C, I, O_f \rangle$ of a row and of an output function consists from a set C of covered rows, a set I of intersected rows and a set O_f of possible function's output values.

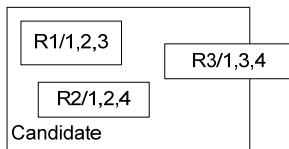


Fig. 15.1. Attribute's interpretation

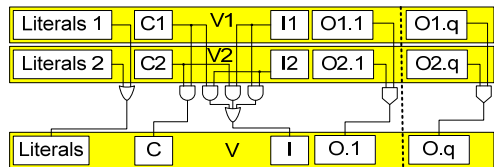


Fig. 15.2. Attribute's computation

If a candidate covers two rows R1 and R2 and intersect row R3, like in Figure 15.2, it is obvious that the only possible output value is 1. The output cannot be 2 because R3 has not this output value on the common zone, it cannot be 3 because R2 has not this output value, it cannot be 4 because R1 has not this output value and it cannot be 5 because R1 and R2 do not have this output value.

The candidates attribute for a function with n outputs is $\langle C, I, \langle O_{f_1}, O_{f_2}, O_{f_3}, \dots, O_{f_n} \rangle \rangle$.

Starting from two candidates $V_1 < C_1, I_1, O_{1f} >$ and $V_2 < C_2, I_2, O_{2f} >$, the attributes of new candidate $V < C, I, O_f >$ are performed using the following formulas:

- $C = C_1 \cap C_2$
- $I = (C_1 \cap I_2) \cup (I_1 \cap C_2) \cup (I_1 \cap I_2)$

The last attribute O_f is performed starting from O_{1f} or O_{2f} . The algorithm use this attribute to compute appropriate output values for the current candidate. An output value is acceptable if belong to any covered or intersected row. The used structure is a table that keep in a descendence order the number of incidences of each output value for covered or intersected rows.

By using this supplementary structure witch keep the set of attributes computed for each candidate, the algorithm avoid redundant computations. In a complex system like the presented one this approach increase with a factor the used memory exponentially decrease the computation.

The candidate's classification: Each candidate is classifiable according to it's attributes. The classification is performed after each computed "child" (component of C). If set C is void, then the candidate is not viable and the algorithm deletes it out of the package. This candidate cannot cover any row and, even by combining it with other candidate, in no way will cover any row. If a candidate with a non void C has available outputs for a function (O_f is not void), it is an implicant for that function and its copy is kept in the set of results. In the package, the attribute O_f is set to "null", and the algorithm will no longer compute the attribute for this function for any child of C . The algorithm keeps candidates if its sets C are not void and all sets O_f are different from the null set.

Overview of implicants generation: All previous presented methods are parts of the general algorithm. After generation of all implicants with the same number of literals, a general structure that keeps state of the algorithm is updated.

Definition: A function for a row is covered if all its output values are covered.

Definition: A row is covered if all its functions are covered.

According to previous definitions, this stage of algorithm is ended when all rows are covered.

Another approach could be performed if the definition of a covered function is changed, i.e. a function for a row is covered if at least one output value of the function is covered. The second approach not preserves

all initial specifications, but, in many cases, generates a satisfactory solution.

The structure keeping the algorithm status could be updated after the generation of all implicants with the same number of literals, or after each implicant generation. In the second case, the algorithm has a higher convergence, but not all implicants with the same number of literals are generated. Such an approach leads to a heuristic method. All these observations are easily to be implemented and have impact in algorithm at the level of candidate classification only.

15.3 Conclusions

The presented method minimizes multiple output multivalent decisional systems, deterministic or nondeterministic, completely or incompletely specified, with correlated or non-correlated outputs. The classical methods of multivalent minimization use the reduction of decision diagrams to the computation of a minimal decisional tree for the given specification. If the given specification is incompletely specified, then the results of this approach include all the input space. Presented method uses only a part of the non-specified space. The scope of this method is not to include unspecified combinations of inputs. The method finds the elements that discriminate input combinations which generate different outputs [9]. The binary specifications become a particular case of the covered class of problems.

The algorithm facilitates the parallel and network minimization. The network minimization automatically establishes an order for the function computation which gives remarkable results even for some classical problems (ex. BCD 7 segments) [10]. The network minimization is applicable on the intrinsic multivalued variable or on the valences of outputs (each output valence corresponds to a subsystem which decides whether or not if the value is delivered to the output). The optimal solution is evaluated after its minimum number of literals. There exist many equivalent solutions with the same number of literals. The method computes and gives as the result only the first of them. If there exists some supplementary criteria that could make the difference between two equivalent solutions, both having a minimum number of literals, (one is better, or other is not acceptable), the algorithm could be adjusted to include the new rule and to reduce the searched space of solutions.

References

1. Quine WV (1952) The problem of simplifying truth functions. *The American Mathematical Monthly*. 59(8):521-531.
2. Brayton RK, Hachtel GD, McMullen C, Sangiovanni-Vincentelli AL (1984) *Logic minimization algorithms for VLSI synthesis*. Kluwer Academic Publisher, Boston
3. Hachtel Gary D, Somenzi Fabio (1984) *Logic synthesis and verification algorithms*. Kluwer Academic Publisher, Boston
4. Sentovich EM, et al. (1992) SIS: A system for sequential circuit synthesis. Univ. of California, Berkeley, CA 94720, Tech. Rep. UCB/ERL M92/41 Electronics Research Lab
5. Gao M, Jiang J-H, Jiang Y, Li Y, Sinha S, Bryton R (2001) MVSIS. In: the Note of the International Workshop on Logic Synthesis, Tahoe City
6. Chai Donald, Jiang Jie-Hong, Jiang Yunjian, Li Yinghua, Mischenko Alan, Brayton Robert (2003) MVSIS 2.0 User's manual. University of Berkeley CA 94720
7. Fišer P, Hlavička J (2003) BOOM - A heuristic boolean minimizer. *Computers and Informatics* 22(1):19-51
8. Fišer Petr, Kubatova Hana (2006) Flexible two-level boolean minimizer BOOM-II and its applications. In: *Proceedings of the 9th EUROMICRO Conference on Digital System Design*, pp 369-376
9. Ștefănescu Ion (2005) DISCRIMINATION: A new principle in the efficient minimization of the binary and multiple-valued functions. In: *1st Int Conf on Electronics, Computers and Artificial Intelligence, ECAI 2005*, University of Pitești, Romania
10. Ștefănescu Ion, Zafiu Adrian (2007) An efficient network strategy in deep minimizations of deterministic and nondeterministic multivalued decisional systems. In: *2nd Int Conf on Electronics, Computers and Artificial Intelligence ECAI 2007*, University of Pitești, Romania

Part 10

Natural phenomena and computer models

This part will discuss the issues and the problems associated with the simulation of natural phenomena. This is a difficult area of research since it generally involves complex data bases and in many instances time variant phenomena. The computational loads can become very high as one considers the physics or the mathematical modeling of structures. Most items in nature, wind, clouds, fire and comets being some examples have not been displayed realistically in computer graphics. This lack stems from a few different problems, all of which are significant. The first is the fact that realistic descriptions require large amounts of storage and consequently large compute time. Nature is able to create diverse detail at the most minute levels within an object of very large scale. The second problem is that of diversity of design within a given framework. Humans typically become tired after the first few iterations of such a design process, with a resulting degradation in the subsequent models. Here we present several results for a multi-layer finite volume solution of wind induced basin flow, NASIR heat generation and transfer solver for RCC dams, water hammer modeling using 2nd order Godunov finite volume method, simulation of wind pressure on two tandem tanks and the tropical cyclone forecaster integrated with case-based reasoning. However, it appears that natural phenomena will require more research into the fundamental way things happen in nature, and in terms of computer graphics that still require new modeling techniques.

Chapter 1

Multi-layer finite volume solution of wind induced basin flow

Saeed-Reza Sabbagh-Yazdi,¹ Hamed Arabi,² Nikos E Mastorakis³

¹ Associate Professor of K.N. Toosi University of Technology, Department of Civil Engineering, 1346 Valiasr St. 19697 Tehran, IRAN
SYazdi@kntu.ac.ir

² MSc. Graduate Student of Science & Research Unite of IA University

³ Professor of Military Insitutes of University Education (ASEI) Hellenic Naval Academy, Terma Chatzikyriakou, 18539, Piraeus, GREECE
mastor@wseas.org

Abstract. In this paper, a multi layers numerical model is introduced for modeling shallow water flows. The model numerically solves equations of continuity and motions in order to compute water depth and velocity patterns in each layer. The model can consider the elevation variation of upper layers and bed and wall geometric complexities and resistances. It can also consider the effect of wind on surface layer. The governing equations are discretized using cell vertex finite volume method in triangular unstructured meshes. For stabilizing the explicit solution process, artificial viscosity formulations are adopted for the unstructured meshes in such a way that preserves the accuracy of numerical results. The accuracy of results of present multi-layer flow solver is assessed by simulating wind induced flow in a circular basin and comparison of computed results with the results of previous research works.

Keywords. Finite Volume Method, Multi-Layer Algorithm, Shallow Water Equations, Wind Induced Flow

1.1 Introduction

Using the hydrostatic pressure assumption and integration of the incompressible continuity and momentum equations end up with two-dimensional depth averaged flow equations known as shallow water equations (SWE).

In some cases of the real world lake flow problems (i.e. shallow lakes with considerable bed roughness which are subject to wind on water surface) vertical variations of in horizontal velocity components are significant. In such cases, although the vertical momentum is mostly due to hydrostatic pressure, consideration of vertical gradients of the horizontal momentum is essential for, particularly for modeling times variation of some parameters like heat, salt and density and other materials which may be solved or suspended in the water. The multilayer solution of SWE is one of the techniques which can be used for numerical solution of such cases.

Some of the works on numerical multi layer simulation of the lakes flow could be referred to Ezar & Mellor in 1993 which used $k-\varepsilon$ turbulence model in multi layer form for analyzing the lakes flow [1]. Mellor et al. in 1998 analyzed the two dimensional model in sigma coordinate considering the effects of advection and diffusion, density and pressure gradients considered [2].

Interaction of wind effects and geometrical features of the lakes (i.e. bathymetry and coastal irregularities) plays important roles in formation of wind induced flow patterns. Therefore, development of a finite volume SWE solver with an algorithm suitable for unstructured meshes is considered in this work.

In this work, the equations of continuity and motions (in horizontal plane) are chosen as the mathematical model which the pressure distribution in flow is hydrostatic and the effect of momentum in vertical direction is negligible. Because of the characteristics of the application problem desired in this work (topography, bed slop and boundary fractures), the sigma coordinate multilayer system is used for numerical modeling. In order to reduce the computational work load, a reconstruction free cell-vertex finite volume algorithm is utilized in the present multi layer SWE solver.

1.2 Governing equations

With assumption of water incompressibility the mathematical multi layer model contains equations of continuity and motions. The continuity equation for the k^{th} layer can be written in vector form as follow:

$$\frac{\partial h_k}{\partial t} + \frac{\partial h_k u_k}{\partial x} + \frac{\partial h_k v_k}{\partial y} = \pm (q_z)_k \quad (1.1)$$

where t is time, x and y are Cartesian coordinates, h is water depth, u and v are velocity components in x and y directions. In this equation, first term shows the changing of flow mass and the second and third terms show the transported mass of flow in x and y directions. q_z is the net balance of water volume exchange in vertical direction between the k^{th} layer and its neighboring media.

The horizontal x and y directions momentum equations for the k^{th} layer could be written as follow:

$$\frac{\partial(h_k v_k)}{\partial t} + \frac{\partial(h_k v_k u_k)}{\partial x} + \frac{\partial(h_k v_k^2)}{\partial y} + g h_k \frac{\partial \zeta}{\partial y} = \frac{\tau_{by}}{\rho} + \left[\frac{\partial}{\partial x} h_k ((T_{yx})_k) + \frac{\partial}{\partial y} h_k ((T_{yy})_k) \right] \quad (1.2)$$

$$\frac{\partial(h_k u_k)}{\partial t} + \frac{\partial(h_k u_k u_k)}{\partial x} + \frac{\partial(h_k u_k v_k)}{\partial y} + g h_k \frac{\partial \zeta}{\partial x} = \frac{\tau_{bx}}{\rho} + \left[\frac{\partial}{\partial x} h_k ((T_{xx})_k) + \frac{\partial}{\partial y} h_k ((T_{xy})_k) \right] \quad (1.3)$$

Here, ζ is depth of each layer from bed (water level in each layer), $\partial(h_k u_k)/\partial t$, $\partial(h_k v_k)/\partial t$ are momentums changing in time of Δt and $\partial(h_k u_k^2)/\partial x$, $\partial(h_k u_k v_k)/\partial y$, $\partial(h_k v_k u_k)/\partial x$, $\partial(h_k v_k^2)/\partial y$ show the momentum effect on flow in x and y directions. $g h_k \partial \zeta / \partial x$, $g h_k \partial \zeta / \partial y$ cite the changing of bed slop and their effect of on flow in x and y directions, and also the effect of upper layer weight on desired layer [3]. τ_{kx} , τ_{ky} representing the x and y directions global stresses of the k^{th} layer, including the effect wind stresses τ_{tx} , τ_{ty} on the upper layer, τ_{bx} , τ_{by} representing the effect of the bed roughness in the lower layer [4].

It worth noting that, negligible shear stress between the layers may be assumed, and, the effects of wind and bed roughness can be interpolated between the momentum equations of all the layers.

The wind stresses that should be added to the above mentioned x and y directions global stresses of the top layer are defined as,

$$\tau_{tx} = \rho_a C_w W_x |W_{10}| \quad (1.4)$$

$$\tau_{ty} = \rho_a C_w W_y |W_{10}| \quad (1.5)$$

where, W_{10} is the wind speed at 10 meters above the water surface and C_w is the drag coefficient which can be obtained from following graph.

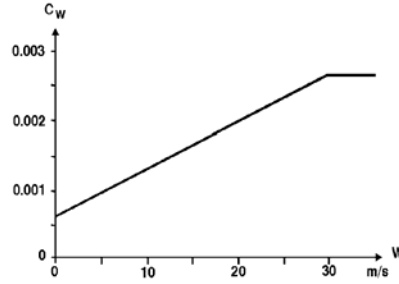


Fig. 2: Relationship between wind speed & drag coefficient [5]

The global friction due to bed roughness can be attained from the following relations:

$$\tau_{bx} = C_f u \sqrt{(u^2 + v^2)}_k \quad (1.6)$$

$$\tau_{by} = C_f v \sqrt{(u^2 + v^2)}_k \quad (1.7)$$

where, C_f is the effective global dissipative coefficient.

In multi layer cases, in which shear stress between the layers are assumed to be negligible (due to uniformity of the fluid properties), the global effect of bed friction that must be added to all the layers may be calculated using following coefficient of friction,

$$C_f = \left[\frac{1}{K} \log \left(\frac{30 Z_b}{k_s} \right) \right]^{-2} \quad (1.8)$$

where, k_s is the bed friction coefficient, k is Von-Karman coefficient which equals to 0.4 and Z_b is distance from bed.

The effect of global dissipation in x and y directions are computed using the following formulation [6].

$$T_{xx} = \frac{\partial}{\partial x} \left(2\nu_t h \frac{\partial u}{\partial x} \right) \quad (1.9)$$

$$T_{xy} = \frac{\partial}{\partial y} \left(v_t h \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right) \quad (1.10)$$

$$T_{yx} = \frac{\partial}{\partial x} \left(v_t h \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) \right) \quad (1.11)$$

$$T_{yy} = \frac{\partial}{\partial y} \left(2v_t h \frac{\partial v}{\partial y} \right) \quad (1.12)$$

where v_t is the horizontal eddy viscosity parameter that can be computed with algebraic, Smagorinsky, $k-\varepsilon$ models. In the present work, the widely used depth-averaged parabolic turbulent model is applied, in which the eddy viscosity parameter is computed algebraic formulation $v_t = \theta h U_*$. In this formulation the bed friction velocity is defined as $U_* = [C_f (u^2 + v^2)]^{0.5}$ and the empirical coefficient θ is advised around 0.1 [7].

1.3 Numerical formulation

Here, the cell vertex finite volume method is applied for converting the governing equations of the layers into discrete form on unstructured triangular meshes. In order to discrete the solution domain into triangular sub-domains Deluaney Triangulation method is applied [8]. In the applied solution algorithm, at the outset, the computational control volumes Ω are formed by gathering the triangles on every node. Then, the governing equations are integrated over each control volume. Application of the Green's theorem to the integrated equations results in:

$$\int_{\Omega} \frac{dW}{dt} d\Omega + \oint_{\Gamma} ((E \Delta y - F \Delta x) + (G \Delta y - H \Delta x)) = \int_{\Omega} S d\Omega \quad (1.13)$$

where,

$$W = \begin{pmatrix} h_k \\ h_k u_k \\ h_k v_k \end{pmatrix}, \quad E = \begin{pmatrix} h_k u_k \\ h_k u_k^2 \\ h_k u_k v_k \end{pmatrix}, \quad F = \begin{pmatrix} h_k v_k \\ h_k u_k v_k \\ h_k v_k^2 \end{pmatrix},$$

$$G = \begin{pmatrix} 0 \\ h_k v_{Th} \frac{\partial u_k}{\partial x} \\ h_k v_{Th} \frac{\partial v_k}{\partial x} \end{pmatrix}, \quad H = \begin{pmatrix} 0 \\ h_k v_{Th} \frac{\partial u_k}{\partial y} \\ h_k v_{Th} \frac{\partial v_k}{\partial y} \end{pmatrix}, \quad S = \begin{pmatrix} q_{zk} \\ -gh_k \frac{\partial \xi}{\partial x} - \frac{\tau_{kx}}{\rho_w} \\ -gh_k \frac{\partial \xi}{\partial y} - \frac{\tau_{ky}}{\rho_w} \end{pmatrix}$$

And Ω and Γ are the area and perimeter of the control volume formed by gathering triangular cells sharing a computational node, respectively.

In order to solve the value of the unknown vector W at the central node of the control volumes, the above integral equation can be transformed to the following algebraic formulation.

$$W_i^{n+1} = W_i^n - \frac{\Delta t}{\Omega_i} \left[\sum_{i=1}^m \left((\bar{E}\Delta y - \bar{F}\Delta x) + (\bar{G}\Delta y - \bar{H}\Delta x) \right) \right]^n + S^n \quad (1.14)$$

where, W_i^{n+1} is the value of W_i^n to be computed after Δt in the i^{th} node.

The parameters $\bar{E}$, $\bar{F}$, $\bar{G}$ and $\bar{H}$ are the averaged values of the fluxes at boundary edges of the control volume Ω [8].

1.4 Artificial Viscosity

In the explicit solution of the convection dominated problems (i.e. flow with negligible friction and turbulent effects) lack of some enough physical dissipation may give rise to some numerical oscillations. This problem, which endangers the stability of the computations, usually appears near regions with high gradient of flow parameters. For damping out these numerical noises artificial dissipation terms can be added to the derived formulations. The formulation of these additional terms should suit unstructured nature of the mesh data. The dissipation terms must damp out the numerical errors without degradation of the accuracy. Although for the flows with gradual variations independent variables the fourth order term (Biharmonic operator) produces enough dissipations, near the high gradient region it is necessary to add second order operators (Laplacian operator) to the dissipation formulation. In order to prevent unwanted dissipation in the smooth flow regions, the Laplacian operator can be multiplied by a water elevation switch. Hence, the artificial dissipation operators are introduced as,

$$(\nabla^4 W_i)_k = \varepsilon_4 \sum_{j=1}^{Ne} [\lambda_{ij} (\nabla^2 W_j - \nabla^2 W_i)]_k \quad \text{and} \quad (\nabla^2 W_i)_k = \varepsilon_2 \sum_{j=1}^{Ne} (W_j - W_i)_k \quad (1.15)$$

Where λ is scaling factor, ε_4 and ε_2 are the artificial dissipation coefficients ($1/256 < \varepsilon_4 < 3/256$ and $0.2 < \varepsilon_2 < 0.3$), which must be tuned to minimum value required for the stability of computations. The scaling factor λ_{ij} is computed using maximum nodal values of Eigen values of Jacobian matrix at the edges connected to the centre node of the control volume. λ is evaluating as follow.

$$\lambda_k = \left[\vec{U} \cdot \hat{n} + \sqrt{U^2 + C^2 (\Delta x^2 + \Delta y^2)} \right]_k \quad \text{where} \quad C = \sqrt{gh_k} \quad (1.16)$$

Where, C is celerity, $\vec{U}$ is averaged computed velocity, and $\hat{n}$ is normal vector at of edges. Note that, the computations of artificial dissipation operators are performed over the edges connected to the central node of the control volume [8].

1.5 Boundary condition

All the inflow and outflow volumes may be considered as source and sinks of the continuity equation at desired nodal points.

Free-slip velocity condition at coastal walls can be imposed where no flow passes through the flow boundaries. These boundaries can be used to reduce the computational domain assuming that the thickness of near wall boundary layer is considerably small in comparison with the horizontal dimensions of the problem. At these boundaries the component of the velocities normal are set to zero. Therefore tangential computed velocities are kept using free slip condition at wall boundaries [8].

1.6 Verification test

Numerical solution of flow in a circular basin is chosen as a test case to evaluate the model performance in simulating the effect of wind flow in a shallow lake.

This test is a circular basin that Krunburg in 1992 design it as shallow basin case. The circular basin has a radius of 192 meter and the depth of 0.6 meter. A steady wind directly passes from east to west of this circle

and its speed is 10 meter per second. The drag coefficient is nearly equal to 0.002 [9].

In present work, the water depth is divided into three layers as surface, middle and bottom layer that it is shown in Figure 1.3.

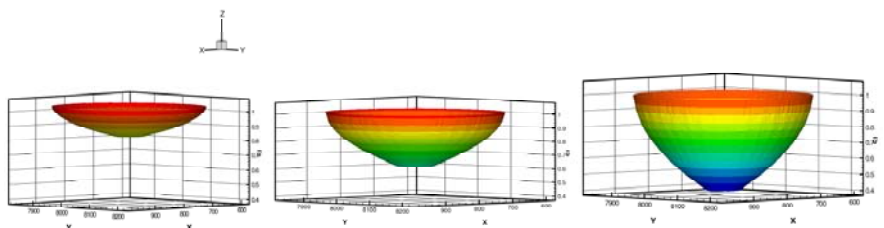


Fig. 1.3. Topology of the three layers of the basin

As there is no internal or external flow through the basin, thereafter just wind stress effects are considered in this case. Still water is assumed as the initial condition. Then, a constant wind with speed of 10 *m/sec* considered to blow 10 meters over the basin. This west to east direction wind formed two circulating flows in the basin. In the following figures the computed flow pattern at each layer are plotted in the form of stream lines (Figure 1.4) and velocity vectors (Figure 1.5).

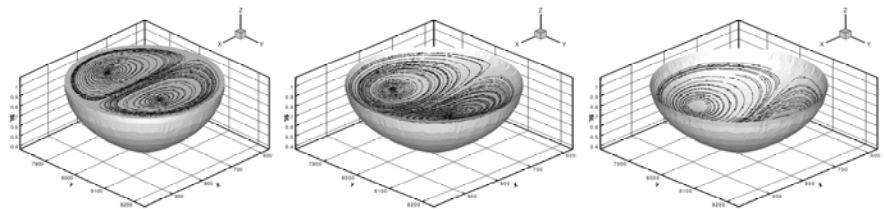


Fig. 1.4. Computed stream lines on surfaces of three layers

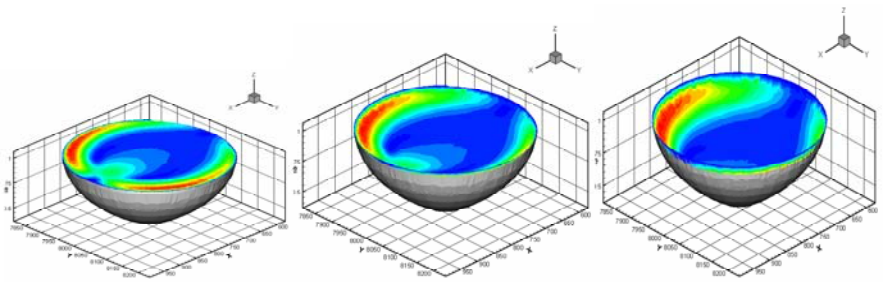


Fig. 1.5. Computed velocity contours on surfaces of three layers

Figure 1.6 presents the results of the present model on unstructured mesh which is very similar to the results reported by previous workers on structured mesh [9].

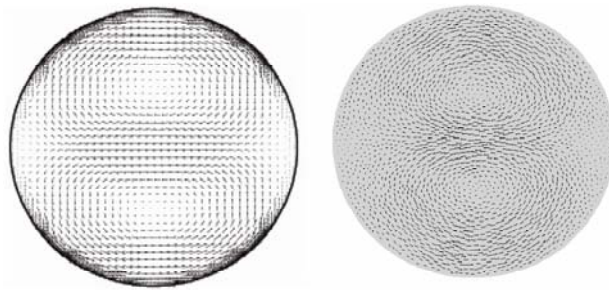


Fig. 1.6. Comparison between velocity vectors of reference paper (up) and results of the present model (down)

1.7 Conclusions

In this work, a multi-layer algorithm for finite volume solution of conservative equations of continuity and motions on unstructured meshes are presented. The multi-layer equations solved on triangular unstructured meshes using cell vertex finite volume method. In this algorithm the effects wind stress is successfully imposed on water surface (top layer) and the hydrostatic pressure and global stresses are distributed over the layers. Oscillation free numerical results are accurately obtained for the flows with wide range variation in depth by application of artificial dissipation operators. In order to validate the results, a circular basin was chosen and the results of present model were compared with the available results of previous reported research works.

References

1. Ezar T, Mellor GL (1993) Model simulated changes in transport. Meridional heat & Costal Sea Level, Department of Physics, Memorial University of New Found land, Canada
2. Mellor GL, Oey LY (1998) American Metrological Society. Journal of Atmospheric & Oceanic Technology 15:11-22
3. User Documentation Coherens, Release 8.4 (1999) A Coupled Hydro dynamical Ecological Model for Regional & Shelf Seas
4. Abbot MB (1980) Computational hyraulic. Pitman Publishing Limited

5. Reference Manual of MIKE 3/21 FM, Estuarine and Coastal Hdraulics and Oceanography, Hydrodynamic Module, DHI Water & Environmental, Ager, Denmark
6. Vreugdenhil CB (1994) Numerical methods for shallow-water flow. Kluwer Academic Publishers
7. Sabbagh-Yazdi SR and MohamadZadeh M (2004)" Finite Volume Solution of Two-Dimensional Convection Dominated Sub-Critical Free Surface Flow Using Unstructured Triangular Meshes " International Journal of Civil Engineering, Vol2, No2, 78-91.
8. Weatherill NP, Hassan O, Marcum DL (1993) Calculation of steady compressible flow-fields with the finite elements method. 31 Aerospace Sciences Meeting and Exhibit, AIAA-93-0341
9. Kranenburg C (1992) Wind-driven chaotic advection in a shallow mode lake. Journal of Hydraulics Research 30(1):29-47

Chapter 2

NASIR heat generation and transfer solver for RCC dams

Saeed-Reza Sabbagh-Yazdi,¹ Ali-Reza Bagheri,² Nikos E Mastorakis³

¹ Associate Professor of K.N. Toosi University of Technology, Department of Civil Engineering, 1346 Valiasr St. 19697 Tehran, IRAN
SYazdi@kntu.ac.ir

² Assistant Professor of K.N. Toosi University of Technology, Department of Civil Engineering, 1346 Valiasr St. 19697 Tehran, IRAN

³ Professor of Military Institutes of University Education (ASEI) Hellenic Naval Academy, Terma Chatzikyriakou, 18539, Piraeus, GREECE

Abstract. In this paper, the heat transfer module of NASIR (Numerical Analyzer for Scientific and Industrial Requirements) software which solves the temperature field on unstructured finite volumes is introduced. In this software, the transient PDE for heat transfer in solid media is coupled to a suitable concrete heat generation algebraic relation. The discrete form of the heat transfer equations is derived by multiplying the governing equation by a piece wise linear test function and integrating over a sub-domain around any computational node. The solution domain is divided into hybrid structured/unstructured triangular elements. The triangular elements in the structured part of the mesh can be activated for simulating gradual movement of the top boundary domain due to advancing the concrete lifts. The accuracy of the developed model is assessed by comparison of the results with available analytical solutions and experimental measurements of two-dimensional heat generation and transfer in a square domain. The computer model is utilized to simulate the transient temperature field in a typical RCC dam.

Keywords. Heat Generation and Transfer, NASIR Finite Volume Solution, RCC Dam Construction Program

2.1 Introduction

The heat exchange in RCC dams principally takes place in the transverse direction of the dam axis. Therefore, two-dimensional models are generally adopted for temperature studies [1,2]. The temperature analysis of a concrete structure is associated with temperature dependent heat sources of cement hydrations all over the domain. However, the heat generation rate of the cement contents with gradually increasing section, depends on the temperature of surrounding concrete and the heat conduction properties of the aggregates, the computation of the temperature profiles is not an easy task.

In this paper, first the mathematical model of the phenomena including the transient partial differential equation of heat transfer and associating boundary conditions in conjunction with an algebraic formula for the source term of the heat generation rate is described. Then, the numerical techniques employed for deriving the algebraic formula suitable for numerical solution on a triangular mesh is discussed. For the present solution algorithm, the domain discretization technique using hybrid unstructured/structured triangular mesh, taking into account the layer-based movement of the top surface boundary of the concrete structure is introduced. The accuracy of the solution of the spatial derivative part of the governing equation is verified using the analytical solution of heat transfer in a square domain. Then, the quality of the formula adopted for the source term of cement heat generation rate is assessed by the use of available experimental measurements on a concrete block. Finally, the developed model is applied for prediction of the transient temperature field in a typical RCC dam section during the construction and after the completion.

2.2 Governing equations

Assuming isotropic thermal properties for the solid materials, the familiar two-dimensional equation defining heat generation and transfer is of the form,

$$\alpha \nabla^2 T + \frac{\alpha}{\kappa} \dot{Q} = \dot{T} \quad (2.1)$$

where, the parameters are $T(^{\circ}C)$ temperature, $\kappa (W/m^{\circ}C)$ heat conduction coefficient and $\dot{Q} (KJ/m^3h)$ rate of heat generation per unit volume and the thermal diffusion is defined as $\alpha = \kappa / \rho C$ with $\rho (Kg/m^3)$ being

the density and C ($KJ/Kg^{\circ}C$) the specific heat of concrete. The parameter $\dot{Q}$, which acts as a source term, is the rate of heat generation due to cement hydration in the concrete field.

The natural boundary condition on concrete external surface is taken as,

$$\kappa(\nabla T \cdot \hat{n}) + q = 0 \quad (2.2)$$

where, q is the rate of heat exchange per unit volume of concrete surface with surrounding ambient and $\hat{n}$ is the unit vector normal to the boundary surface. The rate of heat exchange q is taken into consideration through three mechanisms, q_c (convection), q_r (long wave radiation of concrete to surroundings) and q_s (solar radiation absorption). Hence, the total rate of heat exchange can be defined as, $q = \pm q_c + q_r - q_s$ [1,2].

Note that, q_c is through air (or water) movement and depends on the difference between temperature of the concrete surface, T_s , with surrounding ambient temperature, T_{air} , and is given by $q_c = h_c(T_s - T_{air})$. Where, $h_c = h_n + h_f$ is coefficient of thermal convection comprising from the natural convection coefficient, $h_n = 6(W/m^2^{\circ}C)$, and forced convection coefficient, $h_f = 3.7V$, where $V(m/s)$ is the wind speed.

For long wave radiation a similar relationship is used as $q_r = h_r(T_s - T_{air})$. Where, h_r is the coefficient of thermal radiation.

Short wave exchange rate q_s is given by $q_s = \alpha I_n$. Where α is surface absorption coefficient and the parameter I_n is incident normal solar radiation.

One of the main factors influencing temperature profiles in concrete sections is the amount and rate of heat generated by the cementitious materials present in the concrete [3]. Due to the considerable influence of concrete temperature on the hydration rate of cementation materials, it is necessary to take this effect into account in the simulation method [1,4,5].

The effect of temperature history of each point on its rate of heat generation was taken into account in this research by the concepts of maturity functions $f(t)$ and equivalent time, t_e . Maturity functions define the temperature dependence of cement hydration reactions. Two commonly used such functions are named after Rastrup [2] and Arrhenius [1,5] and are given as:

$$(\text{Rastrup}) f(T) = k 2^{\left(\frac{T}{10}\right)} \quad (2.3)$$

$$(\text{Arrhenius}) f(t) = k \exp \left\{ \frac{-E}{R(273 + T)} \right\} \quad (2.4)$$

K = Constant factor

T = Concrete temperature in $^{\circ}\text{C}$

E = Activation Energy (J/mol)

$E = 33500$ for $T \geq 20^{\circ}\text{C}$

$E = 33500 + 1470(20 - T)$ for $T < 20^{\circ}\text{C}$

R = Molar gas constant = $8.314 \text{ J/}^{\circ}\text{C mol}$

As seen above, an increase in concrete temperature, T , results in an increase in the maturity function, $f(t)$. Using such functions, the relative speed of reaction at any temperature, T , in comparison with that at a reference temperature, T_r , can be obtained. The relative speeds of reaction, $H(T)$, based on Rastrup and Arrhenins functions are derived as

$$(\text{Rastrup}) H(T) = 2^{0.1(T - T_r)} \quad (2.5)$$

$$(\text{Arrhenius}) H(T) = \exp \left\{ \frac{E}{R} \left(\frac{1}{273 + T_r} - \frac{1}{273 + T} \right) \right\} \quad (2.6)$$

Having obtained the relative speed of reaction at various temperatures to that at a reference temperature T_r , the equivalent time (t_e) at this reference temperature for each point in concrete with its particular temperature profile $T(t)$, can be calculated:

$$t_e = \int H(T) dt \quad (2.7)$$

By obtaining the equivalent time, t_e , for each point in concrete, its rate of heat generation can be derived from the base heat generation curve of the concrete under a reference temperature, T_r . The basic heat generation curve is generally derived for the concrete mix under study by experimental determination.

In this work, suitable functions are fitted to the experimental data and the heat generation function $Q(t_e)$ is obtained. Different concretes have

different heat generation characteristics. Therefore, there have been a number of functions proposed for defining this parameter.

Two of the common ones are proposed by Rastrup [2] and Gotfredson [5]:

$$(\text{Rastrup}) Q(t_e) = A + E \exp\{-b[t_e]^n\} \quad (2.8)$$

$$(\text{Gotfredson}) Q(t_e) = A(1 - e^{-\frac{t_e}{B}}) \quad (2.9)$$

where, A, E, B and b are constants determined by regression analysis of experimental results.

Based on the derivations given above the rate of heat generation for any point in concrete can now be determined at any equivalent time, t_e , using the following procedure:

$$\dot{Q}(t_e) = \left(\frac{dQ(t_e)}{d(t_e)} \right) \left(\frac{d(t_e)}{dt} \right) \quad (2.10)$$

Remember that from equation (2.7) we have,

$$\frac{d(t_e)}{dt} = H(t) \quad (2.11)$$

Choosing Rastrups relative rate of reaction for $H(T)$, i.e: equation (2.5) we get:

$$\frac{d(t_e)}{dt} = 2^{0.1(T-T_r)} \quad (2.12)$$

For the term $dQ(t_e)/d(t_e)$, if we choose Rastrups function for basic heat generation curve $Q(t_e)$, we will get:

$$\frac{dQ(t_e)}{d(t_e)} = nbE \{(t_e)^{-n-1} \exp[-b(t_e)^{-n}]\} \quad (2.13)$$

Combining equation (2.12) and (2.13) for obtaining $\dot{Q}(t_e)$ we have:

$$\dot{Q}(t_e) = nbE(t_e)^{-n-1} \exp\{-b(t_e)^{-n}\} 2^{0.1(T-T_r)} \quad (2.14)$$

However, the program enables the user to choose any other combination of maturity function and basic heat generation functions for the derivation of $\dot{Q}(t_e)$.

2.3 Numerical solutions

Here, a fast and accurate numerical technique is introduced which efficiently enables solution of the temperature field in two-dimensional domains with complex and moving boundaries by the use of mixed structured/unstructured triangular meshes [9].

First the governing equation for heat generation and transfer is written in the following two-dimensional form,

$$(i=1,2) \quad \frac{\partial T}{\partial t} + \frac{\partial F_i^T}{\partial x_i} = S \quad (2.15)$$

where T (temperature) is the unknown parameter and S_n is the heat source, and temperature flux in i direction is defined as $F_i^T = \alpha \frac{\partial T}{\partial x_i}$.

The governing equation is multiplied by a piece wise linear test function on triangular element meshes, and then, it is integrated over all triangles surrounded every computational nodes. By application of Gauss divergence theorem and using the property of the test function the boundary integral terms can be omitted. After some manipulations, the resulted equation can be written as,

$$(i=1,2) \quad \int_{\Omega} \left(\frac{\partial T_n}{\partial t} \right) \phi d\Omega = \int_{\Omega} \left(F_i^d \frac{\partial \phi}{\partial x_i} \right)_n d\Omega + \int_{\Omega} S_n \phi d\Omega \quad (2.16)$$

The procedure of deriving algebraic equation will end up with an efficient explicit numerical algorithm by using of following formula within each sub-domain Ω consisting the triangles associated with every node n [9],

$$(i=1,2) \quad T_n^{t+\Delta t} = T_n^t + \Delta t \left[S_n - \frac{3}{2\Omega_n} \left(\sum_{k=1}^N F_i^d \Delta l_i \right)_n \right] \quad (2.17)$$

where, Δl_i is the i direction component of normal vector of m^{th} edge of each triangle. The area of sub-domain, Ω can be computed by summation

of the area of the triangles Λ associated with node n , using $\Lambda = \int_{\Lambda} x_i (d\Lambda)_i \approx \sum_k^3 [\bar{x}_i \delta \ell_i]_k$.

By application of the Gauss divergence theorem, the piece wise constant temperature flux in i direction, F_i^d , at each triangular element can be calculated using following algebraic formula,

$$F_i^d = \frac{1}{\Lambda} \sum_{m=1}^3 (\bar{T} \Delta l_i)_m \quad (2.18)$$

where, $\bar{T}$ is the average temperature of the edge.

Using meshes with various sub-domain sizes of sub-domains, to maintain the stability of the explicit time stepping the minimum time step of the domain of interest should be as,

$$\Delta t = k \left(\frac{\Omega_n}{\alpha_n} \right)_{\min} \quad (2.19)$$

Remember, the heat source for each node n in concrete body is defined by $S_n = \alpha_n \dot{Q}(t_e)_n / \kappa_n$.

If a and κ at node n are considered independent of time and temperature and maturity of the concrete, then for determination of the heat source in every time step, only the value of heat generation rate $\dot{Q}(t_e)$ at nodes located in concrete body must be updated for the nodes using equation (2.3). It should be noted that the value of $\dot{Q}(t_e)$ in each node is a function of t_e for that node. Therefore, t_e should be updated at each time step using following formula,

$$t_e = \sum_{t=t_0}^{t=t_0+N\Delta t} (2^{0.1(T-T_r)}) \Delta t \quad (2.20)$$

Two types of boundary conditions are usually applied in this numerical modeling. The essential and natural boundary conditions are used for temperature at boundary nodes and temperature diffusive flux (gradients) at boundary elements, respectively [10-12].

For the boundaries where the natural boundary conditions are to be applied, by the use of equation (2.2) and the temperature at the nodes at the vicinity of the boundary, the temperature at boundary nodes can be computed and imposed.

2.4 Domain discretization

The solution comprises of the concrete dam body and the foundation (Figure 2.1). Considering the regular geometry of dam cross section and in order to facilitate the movement of the upper boundary of concrete as layered sequences of construction progresses, the use of structured mesh [13] is considered for dam body. Numbering the nodes and elements from the base layer to the top layer of the dam, provides the ability to ignore the desired upper layers of the dam.

Since the age of concrete varies from layer to layer in the body of RCC dam, the use of regular mesh spacing allows moving the top boundary by gradual increasing the number of layers with constant thickness. The structured mesh is therefore quite appropriate because of offering control on the layer thickness and sequence of construction.

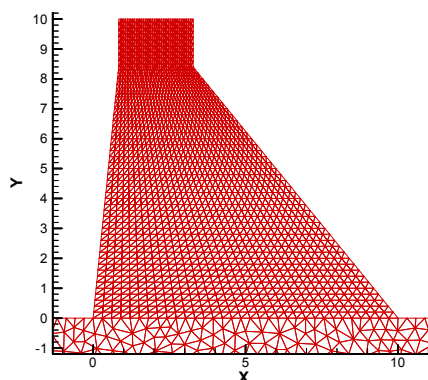


Fig. 2.1. Hybrid structured/unstructured triangular mesh of dam and foundation

The dam foundation, where moving boundary is not required, is discretized using unstructured mesh. This will facilitate consideration of foundations irregularities. The irregular triangular mesh was produced using Deluaney Triangulation technique [13]. The use of unstructured mesh for the foundation of the dam has the advantage of allowing finer mesh size near high heat temperature gradient zones (i.e. the dam body). This technique increases the speed and accuracy of the computations.

2.5 Verification of solution

In order to evaluate the accuracy of the source term representing the rate of the heat generation, a set of experimental measurements on a concrete cube

with 60 (cm) dimensions caste by 450 kilogram per cubic meter is used. The concrete block was insolated all over the faces. The properties of concrete is considered as $\alpha_c = 0.0038 \text{ (m}^2/\text{h)}$ and $\kappa = 9(\text{KJ}/\text{m}^{\circ}\text{ch})$. The averages of measured temperature at three points of this concrete block (one point at the center and two points near the faces) reported by previous workers [7].

A two dimensional rectangular triangular mesh with 6 (cm) spacing is utilized for numerical simulations of this case. The results of the computer model with the 13.5°C concrete placing temperature present reasonable agreements with the experimental data for the permanently isolated concrete block (Figure 2.2).

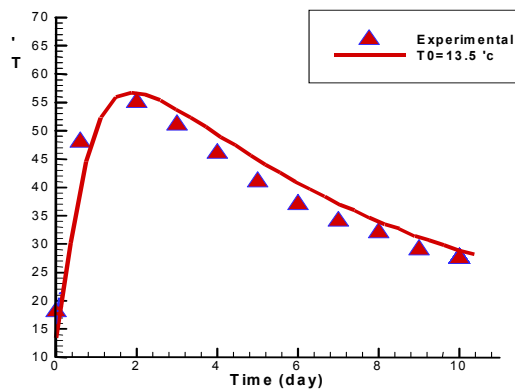


Fig. 2.2. Comparison of computed results ($T_0=13.5^{\circ}\text{C}$) with the experimental data

At the next stage, the accuracy of the solution of spatial derivative terms is investigated by comparison of the results of the numerical solver with the analytical solution of the following two-dimensional boundary value problem with a constant source term as,

$$\Omega = \{0 < (x_1, x_2) < 1\} \quad \frac{\partial^2 T}{\partial x_i^2} = 1 \quad (2.21)$$

The boundary conditions are considered $T = 0$ at $x_1 = 1$, $x_2 = 1$ and $\frac{\partial T}{\partial n} = 0$ at $x_1 = 0$, $x_2 = 0$.

The numerical modeling of the case is performed on 20×20 grid for triangular meshing. In Figure 2.3, the results of numerical computations are compared with the analytical solution of the case [11].

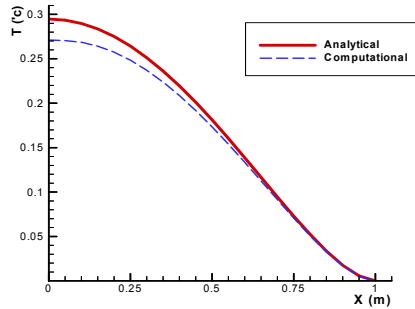


Fig. 2.3. Comparison of computed and analytical results on diagonal line of square

2.6 Applications to a typical case

In order to present the ability of the model to deal with real world cases, it is applied to a typical gravity dam section. Both the base and the height of the section are considered as 10 (m) while the dam crest width is considered as 2.5 (m). Upstream and downstream slopes are designed as 0.1:1.0 and 0.8:1.0, respectively. The radius of the half circle shape far field boundary was considered 1.5 times of the dam base dimension.

The regular mesh of the RCC dam body is generated with 0.2 (m) vertical spacing. The horizontal spacing at the dam base is considered equal to 0.3 (m) and reduced in upper layers proportional to the dam width. The horizontal mesh spacing at the dam base was taken as the finest part of the unstructured triangular mesh of the foundation part of the domain, and the triangles sizes increase far from the dam base center.

Following assumptions were made for concrete placing program. The layers are constructed every 24 hours at the thickness of 0.4 (m). The cement content of the concrete is 150 (kg/m^3). The concrete placing temperature is considered 25 °C. The ambient temperature variation is assumed to vary between 20 and 30 °C. The properties of RCC dam is considered as $\alpha = 0.0038$ (m^2/h) and $\kappa = 9$ ($\text{KJ}/\text{m}^\circ\text{C h}$). Rock foundation is considered to be quartzite with material properties of $\alpha = 0.009$ (m^2/h) and $\kappa = 6$ ($\text{KJ}/\text{m}^\circ\text{C h}$).

The results of the computational model for some of the stages of layer-based construction of a 10 (m) height RCC dam are shown in the form of temperature contour maps at Figures 2.4.

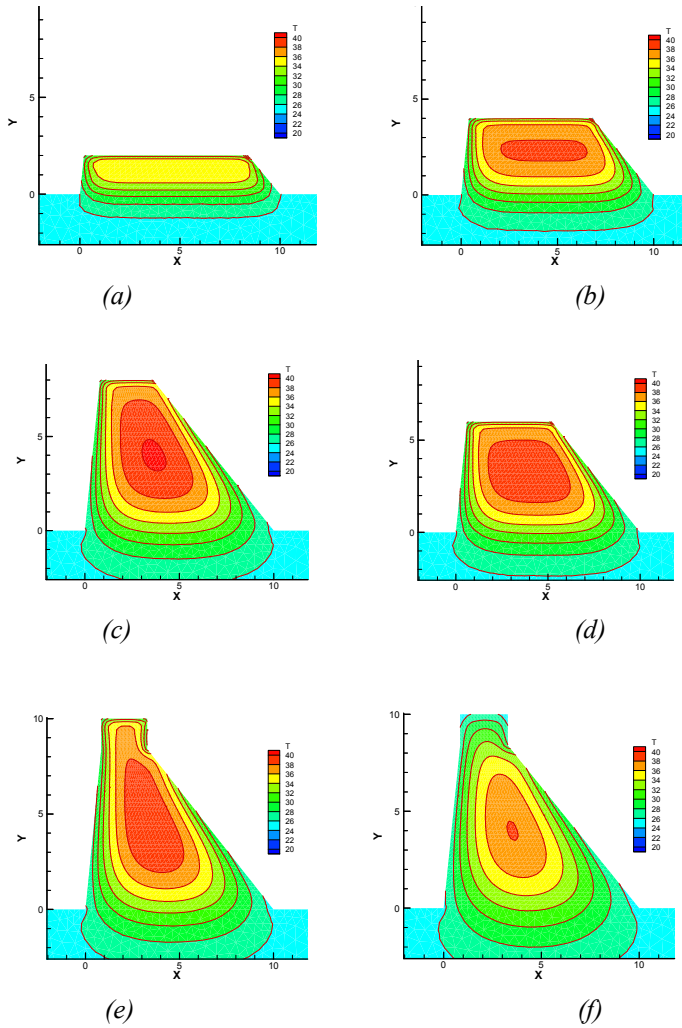


Fig. 2.4. Snap shots from temperature contours at five stages of layered concrete placing of a 10 m height RCC dam without form works during 25 days and finally 10 days after dam completion.

2.7 Conclusions

A module of NASIR software which solves the equation of heat transfer considering the heat generation rate (due to cement hydration) is introduced in this paper. The governing equation is discretized on hybrid struc-

tured-unstructured mesh of triangular element. The resulted algorithm provides light explicit computation of time dependent problems.

The numerical model was verified in two stages. Firstly, the adopted formulation for the term representing heat generation rate of the concrete is evaluated by the comparison of the computed results with the available experimental measurements on a concrete block. Secondly, the accuracy of the solution of the heat transfer terms was assessed using a boundary value problem and its analytical solution. The results of the developed model present reasonable agreements to the analytical solution and experimental measurements.

By application of structured mesh for the concrete dam part of the domain, an efficient modeling of the layered based construction of concrete structures (RCC Dam) is achieved. Gradual movement of the top boundary of the structure during the construction period is simulated by deactivating the upper part of the dam, in which no concrete is placed. For discretization of the dam foundation, unstructured mesh of triangles is considered, which facilitates considering irregularities in geometrical features and material properties of the natural foundation. The developed model was applied to a typical RCC dam section and the results of the temperature fields obtained showed the general pattern expected for such structures.

References

1. Tatro SB, Shrade EK (1992) Thermal analysis for RCC, a practical approach. In: Proceeding of the Conference: Roller Compacted Concrete 3, A.S.C.E., New York
2. Hansen KD, Reinhardt WG (1991) Roller compacted concrete dams. Mac Grow Hill Company, New York
3. Gotfredson HH, Idorn GM (1986) Curing technology at the force Bridge Denmark. ACI S.P. 95:17-33
4. Bagheri AR (1990) Early age thermal effects in conventional and micro-silica concrete linings. PhD Thesis, University of Newcastle, UK
5. Wilson EL (1968) Determination of temperatures within mass concrete structure. Report No. 68-17, Department of Civil Eng., University of California, Berkeley
6. Lachemi M, Aitcin PC (1997) Influence of ambient and fresh concrete temperatures on maximum temperature and thermal gradient in a high performance concrete. ACI Materials Journal, pp 102-110
7. Branco FA, Mendes PA, Mirambell E (1992) Heat of hydration effects in concrete structures. ACI Materials Journal, pp 139-145
8. Sykes LA (1990) Development of a two-dimensional navier-stokes algorithm for unstructured triangular grids. ARA Report 80

9. Sabbagh Yazdi SR, Bagheri AR (2001) Thermal numerical simulation of laminar construction of RCC dams. *Moving Boundaries VI*, WIT Press, pp 183-192
10. Patankar SV (1980) *Numerical heat transfer and fluid flow*. McGraw Hill
11. Reddy JN (1984) *An introduction to the Finite Element Method*. McGraw-Hill, Mathematics and Statistics Series
12. Anderson DA, Tannehill JC, Pletcher RH (1984) *Computational fluid mechanics and heat transfer*. Cambridge Hemispher Press
13. Thompson JF, Soni BK, Weatherill NP (1999) *Hand book of grid generation*. CRC Press.

Chapter 3

Water hammer modeling using 2nd order Godunov finite volume method

Saeed-Reza Sabbagh-Yazdi,¹ Ali Abbasi,² Nikos E Mastorakis³

¹ Associate Professor of K.N. Toosi University of Technology, Department of Civil Engineering, 1346 Valiasr St. 19697 Tehran, IRAN, SYazdi@kntu.ac.ir

² M.Sc Graduate of K.N. Toosi University of Technology, Department of Civil Engineering, 1346 Valiasr St. 19697 Tehran, IRAN, AliAbbasi.civileng@gmail.com

³ Military Institutes of University Education (ASEI) Hellenic Naval Academy, Terma Chatzikyriakou, 18539, Piraeus, GREECE, mastor@wseas.org

Abstract. Analyzing and interpreting water-hammer (unsteady flow) phenomena in a pipeline are not easy tasks. For complicated cases, the governing partial differential equations can only be solved numerically. Various numerical approaches have been introduced for pipe line transient calculations. In this paper, second-order Finite Volume (FV) Godunov type scheme is applied for water hammer problems and the results are analyzed. The developed one-dimensional model is based on Reimann solution. The model is applied to two classic problems (systems consisting of a reservoir, a pipe and a valve). The stability and accuracy of the developed second-order Godunov scheme is examined for Courant number less than or equal to unity. The pressure waves computed by present 2nd order Godunov finite volume method are in close agreement with analytical solution for water hammer problem in a frictionless pipe.

Keywords. Water-Hammer, Unsteady Pipe Flow, Finite Volume Method, Second Order Godunov Type Reiman Solver

3.1 Introduction

The wave resulting from instantaneous and complete valve-closure in pressurized pipe systems propagates upstream and/or downstream in the hydraulic system. The pressure (head) and velocity of the flow (waves) are important parameters in the design of pipeline systems. Thus accurate modeling of water hammer events (hydraulic transient) is vital for proper design and safe operation of pressurized pipeline systems. The design of pipeline systems, and the prediction of water quality impacts, requires efficient mathematical models capable of accurately solving water hammer problems [1].

Various numerical approaches have been introduced for pipeline transient calculation. They include the method of characteristics (MOC), finite difference (FD), wave plan (WP), finite volume (FV), and finite element (FE). Among these methods, MOC proved to be the most popular among water hammer experts. The MOC approach transforms the water hammer partial differential equations into ordinary differential equations along characteristic lines. The integration of these ordinary differential equations from one time step to the next requires that the value of the head and flow at the foot of each characteristic line be known. This requirement can be met by one of two approaches: (i) use the MOC-grid scheme; or (ii) use the fixed-grid MOC scheme and employ interpolation in pipes. However, unwanted numerical damping may associate with application of MOC when the Courant number is less than one. Therefore, application of MOC may end up with accuracy degradation due to problem of making the Courant number exactly equal to one in all the grid points along the pipe lines. [2].

Results of solving the water hammer equations by the explicit and implicit FD schemes show that these second-order FD schemes produce better results than the first-order MOC [1,2].

Finite element methods (FE) are noted for their ability to: (i) use unstructured grids (meshes), (ii) provide convergence and accurate results, and (iii) provide results in any point of problem domain. Jovic (1995) used the combined method of MOC and FE for water hammer modeling in a classical system (a system consisting of a reservoir, a pipe, and a valve) [3].

FV methods are widely used in the solutions of hyperbolic systems, such as gas dynamics and shallow water waves. FV methods are noted for their ability to: (i) conserve mass and momentum, (ii) provide sharp resolution of discontinuities without spurious oscillations, and (iii) use unstructured grid (mesh). The first application of the FV method to water hammer

problems was a first-order scheme which was highly similar to MOC with linear space-line interpolation [4]. Application of second order Godunov scheme for the FV solution of continuity and momentum equations, in which the convective term is ignored, produced accurate results for very low Mach numbers [5].

The objectives of this article are: (1) to implement the Godunov type solution for water hammer and (2) to investigate the accuracy of this method for solution of water hammer problem in a smooth pipe in which there is no physical friction for damping out the numerical errors and oscillations.

3.2 Governing equations

Unsteady closed conduit flow is often represented by a set of 1D hyperbolic partial differential equations [6]:

$$\text{Continuity equation } \frac{\partial H}{\partial t} + V \frac{\partial H}{\partial x} + \frac{a^2}{g} \frac{\partial V}{\partial x} + V \sin \theta = 0 \quad (3.1)$$

$$\text{Momentum equation } \frac{\partial V}{\partial t} + V \frac{\partial V}{\partial x} + g \frac{\partial H}{\partial x} + J = 0 \quad (3.2)$$

where t =time; x =distance along the pipe centerline; $H=H(x,t)$ = piezometric head; $V=V(x,t)$ =instantaneous average fluid velocity; g =gravitational acceleration; a =wavespeed; θ =the pipe slope; and J =friction force at the pipe wall.

The nonlinear convective terms $V \frac{\partial H}{\partial x}$ and $V \frac{\partial V}{\partial x}$ are included in Eqs. (3.1) and (3.2). These terms, although small for the majority of water hammer problems, are not neglected in this paper. Maintaining the convective terms in the governing equations makes the scheme applicable to a wide range of transient flow problems.

3.3 Formulation of finite volume schemes for water hammer

The computational grid involves the discretization of the x axis into reaches each of which has a length Δx and the t axis into intervals each of which has a duration Δt . Node (i,n) denotes the point with coordinate $x=[i-(1/2)]\Delta x$ and $t=n\Delta t$. A quantity with a subscript i and a super-

script n signifies that this quantity is evaluated at node (i, n) . The i th control volume is centered at node i and extends from $i-1/2$ to $i+1/2$. That is, the i th control volume is defined by the interval $[(i-1)\Delta x, i\Delta x]$. The boundary between control volume i and control volume $i+1$ has a coordinate $i\Delta x$ and is referred to either as a control surface or a cell interface. Quantities at a cell interface are identified by subscript such as $i-1/2$ and $i+1/2$.

The Riemann-based FV solution of Eqs. (3.3) and (3.4) in the i th control volume entails the following steps: (1) the governing equations are rewritten in control volume form; (2) the fluxes at a control surface are approximated using the exact solution of the Riemann problems; and (3) a time integration to advance the solution from n to $n+1$.

Eqs. (3.1) and (3.2) can be rewritten in a matrix form as follows:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{A} \frac{\partial \mathbf{u}}{\partial x} = \mathbf{s} \quad (3.3)$$

$$\text{where } \mathbf{u} = \begin{pmatrix} H \\ V \end{pmatrix}; \mathbf{A} = \begin{pmatrix} V & a^2/g \\ g & V \end{pmatrix}; \text{ and } \mathbf{s} = \begin{pmatrix} -\sin \theta \\ -J \end{pmatrix}.$$

For hyperbolic systems in non-conservative form, Eq. (3.3) can be approximated as follows [7]:

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u})}{\partial x} = \mathbf{s}(\mathbf{u}) \quad (3.4)$$

$$\text{where } \mathbf{f}(\mathbf{u}) = \bar{\mathbf{A}}\mathbf{u}; \bar{\mathbf{A}} = \begin{pmatrix} \bar{V} & a^2/g \\ g & \bar{V} \end{pmatrix}; \text{ and } \bar{V} = \text{mean value of } V \text{ to be speci-}$$

fied later. Setting $\bar{V} = 0$, the scheme reverts to the classical water hammer case where the convective terms are neglected.

The mass and momentum equations for control volume i is obtained by integration Eq. (3.4) with respect to x from control surface $i-1/2$ to control surface $i+1/2$. The results is:

$$\frac{d}{dt} \int_{i-1/2}^{i+1/2} \mathbf{u} dx + \mathbf{f}_{i+1/2} - \mathbf{f}_{i-1/2} = \int_{i-1/2}^{i+1/2} \mathbf{s} dx \quad (3.5)$$

Eq. (3.5) is the statement of laws of mass and momentum conservation for the i th control volume. Let U_i = mean value of $\mathbf{u}$ in the interval $[i-1/2, i+1/2]$. Eq. (3.5) becomes

$$\frac{dU}{dt} = \frac{\mathbf{f}_{i-1/2} - \mathbf{f}_{i+1/2}}{\Delta x} + \frac{1}{\Delta x} \int_{i-1/2}^{i+1/2} \mathbf{s} dx \quad (3.6)$$

The fluxes at cell interfaces can be determined from the Godunov schemes that requires the exact solution of the Riemann problem. Godunov schemes are conservative, explicit, and efficient. The formulation of a Godunov scheme for the mass and momentum flux $f_{i+1/2}$ in Eq. (3.6) for all i and for $t \in [t^n, t^{n+1}]$ requires the exact solution of the following Riemann problem:

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u})}{\partial x} = 0 \quad \text{and} \quad \mathbf{u}^n(x) = \begin{cases} \mathbf{U}_L^n & \text{for } x < x_{i+1/2} \\ \mathbf{U}_R^n & \text{for } x > x_{i+1/2} \end{cases} \quad (3.7)$$

where $\mathbf{U}_L^n$ = average value of $\mathbf{u}$ to the left of interface $i+1/2$ at n ; and $\mathbf{U}_R^n$ = average value of $\mathbf{u}$ to the right of interface $i+1/2$ at n . The exact solution of Eq. (3.7) at $i+1/2$ for all internal nodes i and for $t \in [t^n, t^{n+1}]$ is as follows:

$$\mathbf{u}_{i+1/2}(t) = \begin{pmatrix} H_{i+1/2} \\ V_{i+1/2} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} (H_L^n + H_R^n) + \frac{a}{g}(V_L^n - V_R^n) \\ (V_L^n + V_R^n) + \frac{g}{a}(H_L^n - H_R^n) \end{pmatrix} = \mathbf{B}\mathbf{U}_L^n + \mathbf{C}\mathbf{U}_R^n \quad (3.8)$$

where $\mathbf{B} = \frac{1}{2} \begin{pmatrix} 1 & a/g \\ g/a & 1 \end{pmatrix}$; and $\mathbf{C} = \frac{1}{2} \begin{pmatrix} 1 & -a/g \\ -g/a & 1 \end{pmatrix}$.

Using Eq. (3.8), the mass and momentum fluxes at $i+1/2$ for all internal nodes and for $t \in [t^n, t^{n+1}]$ are as follows:

$$f_{i+1/2} = \bar{A}_{i+1/2} \mathbf{u}_{i+1/2} = \bar{A}_{i+1/2} \mathbf{B}\mathbf{U}_L^n + \bar{A}_{i+1/2} \mathbf{C}\mathbf{U}_R^n \quad (3.9)$$

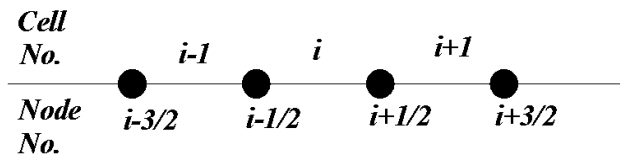


Fig. 3.1. Finite Volume grids system [7]

The evaluation of the right-hand side of Eq. (3.9) requires that $\bar{A}_{i+1/2}$, $\mathbf{U}_L^n$, and $\mathbf{U}_R^n$ are approximated. To estimate $\bar{A}_{i+1/2}$, the entry associated with the advective terms, $\bar{V}_{i+1/2}$, needs to be approximated. Setting $\bar{V} = 0$ is equivalent to neglecting the advective terms from the governing

equations. In general, an arithmetic mean be used to evaluate $\bar{V}_{i+1/2}$ [i.e., $\bar{V}_{i+1/2} = 0.5(V_i^n + V_{i+1}^n)$].

In general, the numerical dissipation in first-order scheme is more than in second-order scheme. Limiters increase the order of accuracy of a scheme while ensuring that results are free of spurious oscillations [4].

Using MINMOD limiter, an approximation for U_L^n and U_R^n that is second order in space and time is obtained as follow:

$$\sigma_{j-1}^n = (U_j^n - U_{j-1}^n) / \Delta x \quad \text{and} \quad \sigma_j^n = (U_{j+1}^n - U_j^n) / \Delta x \quad (3.10)$$

$$\text{MINMOD}(\sigma_j^n, \sigma_{j-1}^n) = \begin{cases} \sigma_j^n & \text{if } |\sigma_j^n| < |\sigma_{j-1}^n| \quad \text{and} \quad \sigma_j^n \cdot \sigma_{j-1}^n > 0 \\ \sigma_{j-1}^n & \text{if } |\sigma_j^n| > |\sigma_{j-1}^n| \quad \text{and} \quad \sigma_j^n \cdot \sigma_{j-1}^n > 0 \\ 0 & \text{if } \sigma_j^n \cdot \sigma_{j-1}^n < 0 \end{cases} \quad (3.11)$$

$$U_{i-(1/2)^+}^n = U_i^n - 0.5\Delta x \text{MINMOD}(\sigma_j^n, \sigma_{j-1}^n) \quad (3.12)$$

$$U_{i-(1/2)^-}^n = U_i^n + 0.5\Delta x \text{MINMOD}(\sigma_j^n, \sigma_{j-1}^n) \quad (3.13)$$

$$U_{i+(1/2)^-}^{n*} = U_{i+(1/2)^-}^n + \frac{1}{2} \frac{\Delta t}{\Delta x} [f(U_{i-(1/2)^+}^n) - f(U_{i+(1/2)^-}^n)] \quad (3.14)$$

$$U_{i-(1/2)^+}^{n*} = U_{i-(1/2)^+}^n + \frac{1}{2} \frac{\Delta t}{\Delta x} [f(U_{i-(1/2)^+}^n) - f(U_{i+(1/2)^-}^n)] \quad (3.15)$$

$$U_R^n = U_{i+(1/2)^+}^{n*} \quad \text{and} \quad U_L^n = U_{i+(1/2)^-}^{n*} \quad (3.16)$$

Inserting Eq. (3.16) into Eq. (3.9) can give Godunov second-order scheme for the water hammer.

3.4 Boundary conditions

The implementation of boundary conditions is a important step in solving partial differential equations. The boundary conditions in this model are [4]:

3.4.1 Upstream head-constant reservoir

The flux at an upstream boundary (i.e., $i=1/2$) can be determined from the Riemann solution. The Riemann invariant associated with the negative characteristic line is:

$$H_{1/2} - \frac{a}{g} V_{1/2} = H_1^n - \frac{a}{g} V_1^n \quad (3.17)$$

Coupling this Riemann invariant with a head-flow boundary relation determines:

$$V_{1/2}^{n+1} = V_{1/2}^n + \frac{g}{a} (H_{1/2} - H_{1/2}^n) \quad (3.18)$$

For an upstream reservoir where $H_{1/2}^n = H_{res}$, the flux at the upstream boundary is:

$$f_{1/2} = \begin{bmatrix} \bar{V}_{1/2} H_{res} + \frac{a^2}{g} (V_1^n + \frac{g}{a} (H_{res} - H_1^n)) \\ g H_{res} + \bar{V}_{1/2} (V_1^n + \frac{g}{a} (H_{res} - H_1^n)) \end{bmatrix} \quad (3.19)$$

3.4.2 Fully closed downstream valve

The flux at a downstream boundary can be determined from the Riemann solution. The Riemann invariant associated with the positive characteristic line is:

$$H_{Nx+1/2} + \frac{a}{g} V_{Nx+1/2} = H_{Nx}^n + \frac{a}{g} V_{Nx}^n \quad (3.20)$$

Downstream boundary condition is valve closure in T_c . Head-flow boundary relation determines:

$$V_{Nx+1/2}^{n+1} = V_{steady} (1 - \frac{t}{T_c}) \quad 0 \leq t \leq T_c \quad (3.21)$$

$$V_{Nx+1/2}^{n+1} = 0 \quad t > T_c \quad (3.22)$$

$$H_{Nx+1/2}^n - H_{Nx-1/2}^n - \frac{a}{g}(V_{Nx+1/2}^{n+1} - V_{Nx-1/2}^n) = 0 \quad (3.23)$$

As a result, the flux at the boundary is determined as follows:

$$f_{Nx+1/2} = \left[\bar{V}_{Nx+1/2} \left(H_{Nx} + \frac{a}{g} V_{Nx}^n - \frac{a}{g} V_{Nx+1/2}^n \right) + \frac{a}{g} V_{Nx+1/2}^n \right] \\ g \left(H_{Nx} + \frac{a}{g} V_{Nx}^n - \frac{a}{g} V_{Nx+1/2}^n \right) + \bar{V}_{Nx+1/2} V_{Nx+1/2}^n \quad (3.24)$$

3.5 Time integration

The previous section provided a second-order scheme for the flux terms. In order to advance the solution from n to $n+1$, Eq. (3.6) needs to be integrated with respect to time. In the absence of friction, the time integration is exact and leads to the following [4]:

$$U_i^{n+1} = U_i^n - \frac{\Delta x}{\Delta t} (f_{i+1/2}^n - f_{i-1/2}^n) \quad (3.25)$$

In the presence of friction, a second order Runge-Kutta solution is used and results in the following explicit procedure:

$$\bar{U}_i^{n+1} = U_i^n - \frac{\Delta x}{\Delta t} (f_{i+1/2}^n - f_{i-1/2}^n) \quad (3.26)$$

$$\bar{\bar{U}}_i^{n+1} = \bar{U}_i^{n+1} + \frac{\Delta t}{2} s(\bar{U}_i^{n+1}) \quad (3.27)$$

$$U_i^{n+1} = \bar{\bar{U}}_i^{n+1} + \Delta t \cdot s(\bar{\bar{U}}_i^{n+1}) \quad (3.28)$$

The time step should satisfy the Courant-Friedrichs-Lewy (CFL) condition for the convective part:

$$Cr = \frac{a \cdot \Delta t}{\Delta x} \leq 1 \quad (3.29)$$

Although another stability condition should be used for the updating of a source term, it is found that the CFL condition is sufficient for the cases where the magnitude of the source term is small.

3.6 Numerical Results and Discussion

The objective of this section is to compare the accuracy and efficiency of Godunov scheme in solving water hammer problems. Analytical solution data [8], and MOC results [9] are used to investigate the accuracy of proposal model (Godunov scheme). The geometrical and hydraulic parameters for this test case are given in Table 3.1. This test case consists of a simple reservoir-pipe-valve configuration.

Table 3.1. Properties for the test case 1

Pipe diameter (m)	0.5
Pipe length (m)	1000
D.W friction factor	0.00
Unsteady friction factor	0.00
Wave speed(m/s)	1000
Reservoir head-upstream (m)	0
Initial mean velocity(m/s)	1.02
Cause of transients	Downstream instantaneous fully valve closure

As can be seen in Figure 3.2, there are minor numerical dissipations produced by the Godunov scheme for Courant number less than one. It worth noting that, the scheme produce pressure waves very similar to the analytical solution when $Cr=1.0$.

Figure 3.3 shows the analytical and computed solutions for the variations in hydraulic head at the valve as a function of time. As expected, the head traces results by both schemes (MOC and FVM) exhibit numerical dissipation, but the numerical dissipation in FVM is less than the MOC, particularly for small Courant numbers (i.e. 0.1).

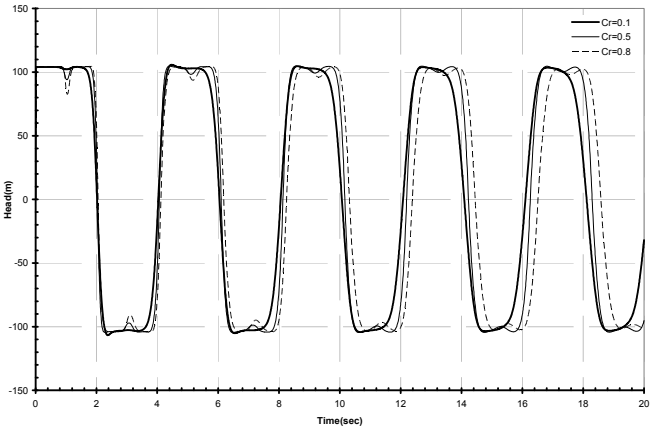


Fig. 3.2. Pressure head traces at valve for various Cr No. (FVM)

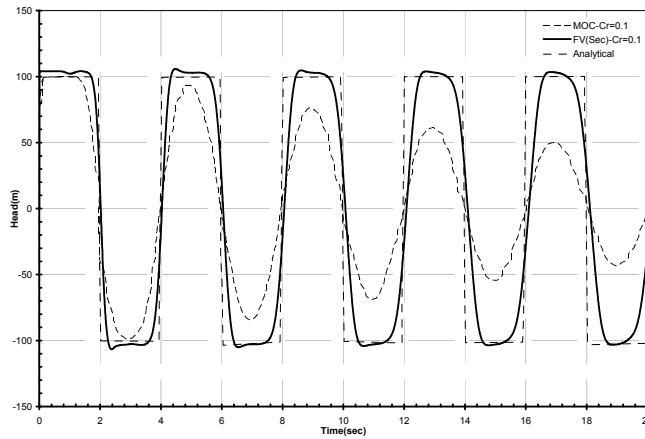


Fig. 3.3. Hydraulic head at the valve for MOC and FVM and analytical solution

3.7 Conclusions

In this paper, nonlinear convective terms are included in governing equations and explicit finite volume formulations are derived using second-order Godunov-type scheme and the model is applied for solution of water hammer problem in smooth pipes. The results of this scheme are compared with analytical solution from different research groups and with numerical data produced by a MOC model. The results are as follows.

1. Finite volume formulation conserves mass and momentum and produce sharp shock fronts.
2. Omission of damping effect of friction due to pipe roughness does not produce any numerical instability which may degrade the solution results
3. The results of explicit Godunov finite volume scheme is in close agreement with analytical solution for the Courant numbers less than one.
4. Numerical dissipation of the explicit Godunov finite volume scheme is considerably less than MOC, particularly for the Courant numbers less than one.

References

1. Wylie EB, Streeter VL (1993) Fluid transients in system. Prentice-Hall, Engle-wood
2. Lai C (1989) Comprehensive method of characteristic models for flow simulation, Journal of Hydraulic Engineering 114(9):1074-1097
3. Jovic V (1995) Finite element and method of characteristics applied to water hammer modeling, International Journal for Engineering Modeling 8(3&4):21-28
4. Guinot V (2000) Riemann solvers for water-hammer simulations by Godunov method. Int J Numer Meth Eng 49:851-870
5. Zhao M, Ghidaoui MS (2004) Godunov-type solution for water hammer flows. J Hydraul Engng 130(4):341-348
6. Chaudhry MH (1979) Applied hydraulic transients. Van Nostrand Reinhold
7. Toro EF (1997) Reimann solvers and numerical methods for fluid dynamics. Springer, Berlin
8. Sharp BB (1981) Water-hammer problems and solutions, University of Melbourne, Edward Arnold Ltd.
9. Turani M (2004) Water hammer analysis with MOC for Karoun IV Penstocks (in Farsi). M.Sc thesis, K.N.Toosi University of Technology

Chapter 4

Simulation of wind pressure on two tandem tanks

Saeed-Reza Sabbagh-Yazdi,¹ Farzad Meysami-Azad,² Nikos E Mastorakis³

¹ Associate Professor of K.N. Toosi University of Technology, Department of Civil Engineering, 1346 Valiasr St. 19697 Tehran, IRAN

SYazdi@kntu.ac.ir

² MSc Graduate of K.N. Toosi University of Technology, Department of Civil Engineering, 1346 Valiasr St. 19697 Tehran, IRAN

³ Proferssor of Military Insitutes of University Education (ASEI) Hellenic Naval Academy, Terma Chatzikyriakou, 18539, Piraues, GREECE

Abstract. In this work, the set of two dimensional for the incompressible fluid is combined with a SGS turbulence viscosity turbulence mode. The SGS turbulence viscosity model is applied to simulate the changes in the pressure distributions in two storage tanks in tandem arrangement due to the wind flow at supercritical Reynolds number (1.43×10^8). These two storage tanks are located in the KAZERUN power station, which are under construction in south part of IRAN. NASIR 2D air flow-solver is applied to investigate the wind pressure on structural surfaces of these tanks. This Numerical model solves the governing equations for wind flow on unstructured finite volumes. The equation of continuity is simultaneously solved with the two equations of motion in a coupled manner for the steady state problems by application of the pseudo compressibility technique. The discrete form of the two-dimensional flow equations are formulated using the Galerkin Finite Volume Method for unstructured mesh of triangular cells. Using unstructured meshes provides the merit of accurate geometrical modeling of the curved boundaries of the tanks.

Keywords. Wind Flow, SGS Turbulent Viscosity Model, Storage Tanks Interference, Triangle Unstructured Mesh, 2D NASIR Finite Volume

4.1 Introduction

The interaction of neighboring tanks may considerably change the pressure filed on a storage tank. Moreover, wind flow through particular arrangement of storage tanks may produce unexpected pressure fields, which may cause disastrous structural loading condition. Therefore, modeling of final design is recommended by most of the codes of practice to evaluate actual pressure loads on them.

The availability of high performance digital computers and development of efficient numerical models techniques have accelerated the use of Computational Fluid Dynamics. The control over properties and behavior of fluid flow and relative parameters are the advantages offered by CFD which make it suitable for the simulation of the applied problems. Consequently, the computer simulation of complicated flow cases has become one of the challenging areas of the research works. In this respect, many attempts have been made to develop several efficient and accurate numerical methods suitable for the complex solution domain.

In the present paper, a computation of pressure distribution of two circular tanks with different diameter in tandem arrangement at supercritical Reynolds number (1.43×10^8) is performed using NASIR (Numerical Analyzer for Scientific and Industrial Requirements) finite volume flow solver and the results are discussed.

4.2 Governing equations

The Navier-Stokes equations for an incompressible fluid combined with a sub grid scale (SGS) turbulence viscosity model are used for the large eddy simulation (LES) of the flow around storage tanks. The non-dimensional form of the governing equations in Cartesian coordinates can be written as:

$$\frac{\partial W}{\partial t} + \left(\frac{\partial F^c}{\partial x} + \frac{\partial G^c}{\partial y} \right) + \left(\frac{\partial F^c}{\partial x} + \frac{\partial G^c}{\partial y} \right) = 0 \quad (4.1)$$

where,

$$W = \begin{pmatrix} \frac{p/\rho_0}{\beta^2} \\ u \\ v \end{pmatrix}, \quad F^c = \begin{pmatrix} u \\ u^2 + p/\rho_0 \\ uv \end{pmatrix}, \quad G^c = \begin{pmatrix} v \\ uv \\ v^2 + p/\rho_0 \end{pmatrix}, \quad F^v = \begin{pmatrix} 0 \\ v_T \frac{\partial u}{\partial x} \\ v_T \frac{\partial v}{\partial x} \end{pmatrix}, \quad G^v = \begin{pmatrix} 0 \\ v_T \frac{\partial u}{\partial y} \\ v_T \frac{\partial v}{\partial y} \end{pmatrix}$$

W represents the conserved variables while, F^c, G^c are the components of convective flux vector and F^v, G^v are the components of viscous flux vector of W in non-dimensional coordinates x and y , respectively, u and v the components of velocity, p pressure are three dependent variables. ν_T is the summation of kinematic viscosity ν and eddy viscosity ν_t .

The variables of above equations are converted to non-dimensional form by dividing x and y by L , a reference length u and v by U_o , upstream wind velocity, and p by $\rho_o U_o^2$.

The parameter β is introduced using the analogy to the speed of sound in equation of state of compressible flow. Application of this pseudo compressible transient term converts the elliptic system of incompressible flow equations into a set of hyperbolic type equations [1]. Ideally, the value of the pseudo compressibility is to be chosen so that the speed of the introduced waves approaches that of the incompressible flow. This, however, introduces a problem of contaminating the accuracy of the numerical algorithm, as well as affecting the stability property. On the other hand, if the pseudo compressibility parameter is chosen such that these waves travel too slowly, then the variation of the pressure field accompanying these waves is very slow. Therefore, a method of controlling the speed of pressure waves is a key to the success of this approach. The theory for the method of pseudo compressibility technique is presented in the literature [2].

Some algorithms have used constant value of pseudo compressibility parameter and some workers have developed sophisticated algorithms for solving mixed incompressible and compressible problems [3]. However, the value of the parameter may be considered as a function of local velocity using following formula proposed [4]

$$\beta^2 = \text{Maximum}(\beta_{\min}^2 \quad \text{or} \quad C|U^2|)$$

In order to prevent numerical difficulties in the region of very small velocities (ie, in the vicinity of stagnation points), the parameter $\beta_{\min}^2$ is considered in the range of 0.1 to 0.3, and optimum C is suggested between 1 and 5 [5].

The method of the pseudo compressibility can also be used to solve unsteady problems. For this propose, by considering additional transient term. Before advancing in time, the pressure must be iterated until a divergence free velocity field is obtained within a desired accuracy. The approach in solving a time-accurate problem has absorbed considerable attentions [6].

In present paper, the primary interest is in developing a method of obtaining steady-state solutions.

4.3 Numerical method

The governing equations can be changed to discrete form for the unstructured meshes by the application of the Galekin Finite Volume Method. This method ends up with the following 2D formulation:

$$\frac{\Delta W}{\Delta t} = \frac{(W_j^{n+1} - W_j^n)}{\Delta t} = -\frac{P}{\Omega} \left[\sum_{k=1}^{Nedge} (F^c \Delta y - G^c \Delta x) \right] - \frac{P}{A} \left[\frac{3}{2} \sum_{i=1}^{Ncell} (F^v \Delta y - G^v \Delta x) \right] \quad (4.2)$$

where, W_i represents conserved variables at the center of control volume Ω_i (Figure 4.1).

Here, F^c, G^c are the mean values of convective fluxes at the control volume boundary faces and F^v, G^v are the mean values of viscous fluxes which are computed at each triangle. Superscripts n and $n+1$ show n th and the $(n+1)$ th computational steps. Δt is the computational step (proportional to the minimum mesh spacing) applied between time stages n and $n+1$. In present study, a three-stage Runge-Kutta scheme is used for stabilizing the computational process by damping high frequency errors, which this in turn, relaxes CFL condition.

In this study, the Smagorinsky model is used for the Sub-Grid Scale (SGS) turbulence viscosity. Eddy viscosity $\nu_t = \nu_{SGS}$ is computed as follow [7]:

$$\nu_{SGS} = (C_s \Delta)^2 [1/2 \bar{s}_{ij} \bar{s}_{ij}]^{1/2} \quad (4.3)$$

$$\approx \frac{1}{\Delta} \sum_1^3 [\bar{\nu} \Delta y - \bar{u} \Delta x] \quad \bar{s}_{ij} = \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \quad (4.4)$$

where, $i, j = 1, 2$ are for the two-dimensional computation in this paper. The Sub-Grid Scale model is used for definition of ν_{SGS} , where Δ is the area of a triangular cell and the $C_s = 0.15$ are used. In equation 4.4, $\bar{u}, \bar{\nu}$ are mean values of velocity in each edge of the triangular element. $\Delta x, \Delta y$ for edge k of control volume Ω (Figure 4.1) are computed as follow:

$$\Delta y_k = y_{n_2} - y_{n_1}, \quad \Delta x_k = x_{n_2} - x_{n_1} \quad (4.5)$$

In order to damp unwanted numerical oscillations associated with the explicit solution of the above algebraic equation a fourth order (Bi-Harmonic) numerical dissipation term is added to the convective, $C(W_i)$ and viscous, $D(W_i)$ terms. Where;

$$C(W_i) = \sum_{k=1}^{N_{edges}} [F^c \Delta y - G^c \Delta x] \quad , \quad D(W_i) = \sum_{k=1}^{N_{cell}} [F^v \Delta y - G^v \Delta x]$$

The numerical dissipation term, is formed by using the Laplacian operator as follow;

$$\nabla^4 W_i = \varepsilon_4 \sum_{j=1}^{Ne} \lambda_{ij} (\nabla^2 W_j - \nabla^2 W_i) \quad (4.6)$$

The Laplacian operator at every node i , is computed using the variables W at two end nodes of all N_{edge} edges (meeting node i).

$$\nabla^2 W_i = \sum_{j=1}^{Ne} (W_j - W_i) \quad (4.7)$$

In equation 4.6, λ_i , the scaling factors of the edges associated with the end nodes i of the edge k . This formulation is adopted using the local maximum value of the spectral radii Jacobian matrix of the governing equations and the size of the mesh spacing as [6]:

$$\lambda_i = \sum_{k=1}^{Ne} \left| (u \Delta y - v \Delta x)_k \right| + \sum_{k=1}^{Ne} \sqrt{(u \Delta y - v \Delta x)_k^2 + (\Delta x^2 + \Delta y^2)_k} \quad (4.8)$$

4.4 Application of the model

According to API 650 storage tanks should remain stable during hurricanes with the speed of 100m/h (44m/s) [8]. for air we have $\rho = 1.23 \text{ kg/m}^3$, $\mu = 1.795 \times 10^{-5}$, So Reynolds number for tank A is computed as;

$$\text{Re} = \frac{\rho v D}{\mu} = \frac{1.23 \times 44 \times 47.5}{1.795 \times 10^{-5}} 1.43 \times 10^8$$

The performance of the this flow-solver is examined by solving turbulence flow around two storage tanks in the KAZERUN power station located in south part of IRAN. In (Figure 4.1) schematic diagram shows the arrangement of two tanks in KAZERUN site.

In order to assess the changes of pressure distribution when two tanks are in tandem arrangement the flow solver is applied to solve the turbulence flow on three different meshes; first tank A, second tank B and finally two tanks with each other (Figure 4.2).

In this work, No-slipping condition is considered at the solid wall nodes by setting zero normal and tangential components of computed velocities at wall nodes. At inflow boundaries unit free stream velocity and at out-flow boundaries unit pressure is imposed. The free stream flow parameters (outflow pressure and inflow velocity) are set at every computational node as initial conditions.

The results on the tank A at supercritical Reynolds number (1.43×10^8) is plotted in terms of velocity vectors respectively (Figure 4.3).

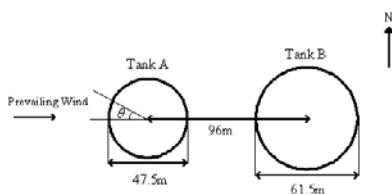


Fig. 4.1. Schematic diagram of the arrangement of two tanks in KAZERUN power plant

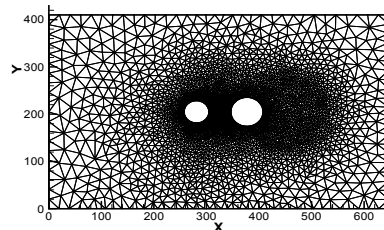


Fig. 4.2. Mesh for two tanks in tandem arrangement

For the case of two storage tanks with tandem arrangement velocity vectors are presented in (Figure 4.4). In this case, two tanks behave as a single lengthened body because the spacing ratio between two tanks is small. As can be seen, there is almost no fluid flow in the gap of between two tanks.

Pressure distributions around tank A, is plotted in (Figure 4.5), Minimum pressure reaches the value of (-2.5) and then raises to (+1). Pressure distribution changes around tank A, where two tanks are in tandem, is illustrated in (Figure 4.6), it can be clearly seen that the pressure distribution on tank A is reduced. Table 4.1 shows the percentage of decrease in pressure coefficient in tank A after addition of tank B.

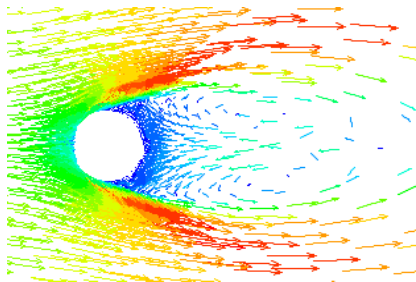


Fig. 4.3. Velocity vectors around a single tank (A) at supercritical Reynolds number (4.52×10^5)

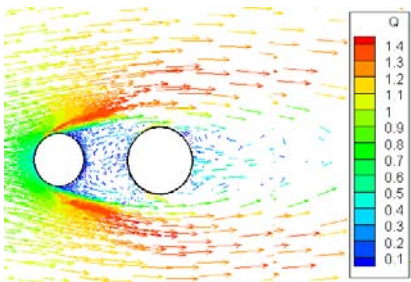


Fig. 4.4. Velocity vectors around two tandem tanks at supercritical Reynolds number (4.52×10^5)

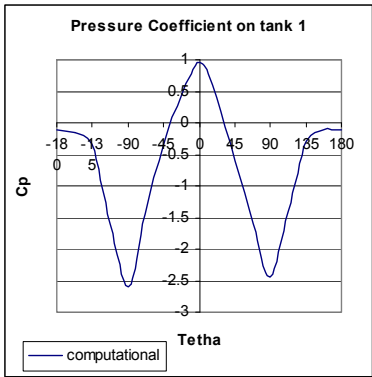


Fig. 4.5. Pressure distributions around tank A (without tank B)

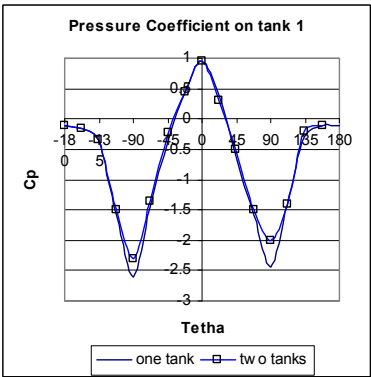


Fig. 4.6. Pressure distribution around tank A (up-stream of tank B)

Table 4.1. Decrease in pressure coefficient on tank A by considering tank B at downstream

θ	$\theta = 0$	$\theta = 90$	$\theta = 180$
Decrease in pressure coefficient	2%	17%	0%

Pressure distributions around tank B is illustrated in (Figure 4.7). Changes in tank B pressure distribution, where two tanks are in tandem, are illustrated in (Figure 4.8), it obvious that the pressure distribution on tank B is reduced highly. Percentage of decrease in pressure coefficient in tank B after addition of tank A is illustrated in Table 4.2. The pressure coefficient decrease up to 90% in $\theta = 0$.

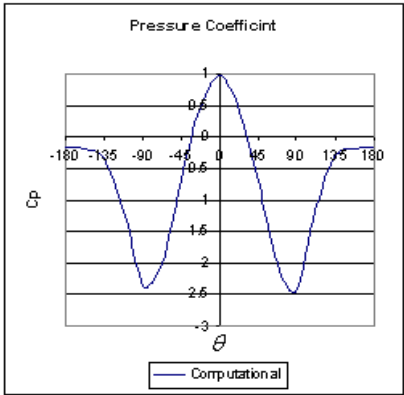


Fig. 4.7. Pressure distributions around tank B (without tank A)

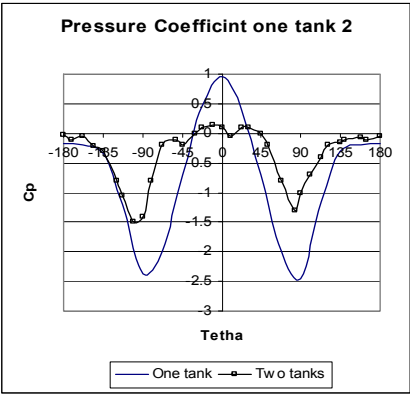


Fig. 4.8. Pressure distribution around tank B (down-stream of tank A)

Table 4.2. Decrease in pressure coefficient on tank B by considering tank A at up-stream

θ	$\theta = 0$	$\theta = 90$	$\theta = 180$
Decrease in pressure coefficient	90%	53%	50%

4.5 Discussion

For the present case of two tanks in tandem arrangement, the downstream tank is submerged in the wake of the up stream one. The computed results show that the interference will be greatly affected by the wake behavior of the up stream tank. When the spacing ratio is small two tanks behave as a single lengthened body. The downstream tank is wrapped in the shear layers separated from the upstream tank. In present case of tandem arrangement, zero computed pressure is detected on a wide area in the front part of the down stream tank.

References

1. Chorin A (1967) A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics* 2:12-26
2. Chang JL, Kwak D (1984) On the method of pseudo compressibility for numerically solving incompressible flow. *AIAA 84-0252, 22nd Aerospace Science Meeting and Exhibition, Reno*

3. Turkel E (1986) Preconditioning methods for solving the incompressible and low speed compressible equations. ICASE Report 86-14
4. Dreyer J (1990) Finite volume solution to the steady incompressible Euler equation on unstructured triangular meshes. M.Sc. Thesis, MAE Dept., Princeton University
5. Rizzi A, Eriksson L (1985) Computation of inviscid incompressible flow with rotation. *Journal of Fluid mechanic*, 153:275-312
6. Belov A, Martinelli L, Jameson A (1995) A new implicit algorithm with multi-grid for unsteady incompressible flow calculations. AIAA 95-0049, 33rd Aerospace Science Meeting and Exhibition, Reno
7. Yu D, Kareem A (1996) Two-dimensional simulation of flow around rectangular prisms. *Journal of Wind Engineering and Industrial Aerodynamics* 62:131-161
8. Lieb JM (2003) API 650 External pressure design appendix. Tank Industry Consultants

Chapter 5

Tropical cyclone forecaster integrated with case-based reasoning

James N.K. Liu, Simon C.K. Shiu, Jane You

Department of Computing, Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
{csnkliu, cscshiu, csyjia}@comp.polyu.edu.hk

Abstract. One of the major challenges for predicting tropical cyclone intensity is that we lack the understanding of coupling relationships of physical processes governing tropical cyclone intensification. This paper presents a Java-based case-based reasoning model to assist tropical cyclone forecasters to determine the intensity change of the tropical cyclone. Cases are constructed by using the data mining algorithms to uncover the hidden relationships between physical processes and tropical cyclone intensity. We specify the domain data, definitions of features from the data, tool for data exploration, and architecture of case-based reasoning model. Preliminary results are found to be useful to forecasters when faced with some unusual problem and under different weather situations.

Keywords. Tropical cyclone forecast, Case-based reasoning (CBR), Data mining, Association rule

5.1 Introduction

Tropical cyclone (TC) forecasting is especially difficult because cyclones are highly chaotic events and do not consistently take the same obvious form or follow obviously established patterns. Cyclones, however, may offer certain consistent information that is not apparent and this information

may be available for classification and further use if suitable techniques can be developed and applied.

The intensity of a TC is one of the important parameters for estimating the possible impact by the TC. The prediction of TC intensity which defined as the maximum sustained 1-minute surface winds remains a challenge task in all tropical cyclone basins. It can be estimated by associating certain cloud features with conventional estimates of its strength. Procedures have been designed to improve both the reliability and the consistency of intensity estimates made from satellite imagery [2]. These procedures rely on the interpretive skill of the meteorologists who manually view the satellite images and make a subjective assessment of the features that may be involved. The image is matched with a number of possible pattern types: curved band pattern, shear pattern, eye pattern, etc [5]. Each TC goes through a life cycle that may be classified into eight categories (T-numbers) by its appearance in visible satellite imageries and the use of a decision tree. However, due to the subjective analysis technique and the rapid development of TCs in severe weather conditions, such prediction always lead to inaccurate results [11].

We have applied data mining techniques to meteorological data lately and uncovered interesting inter-transactional association rules for weather prediction [4]. This represents a step forward for the association of explanation with the weather phenomena. The aim of this study is to develop some means of extracting cases based on the previous meteorological information and important signals and implements a system for forecasting the intensity of a TC. We note that Case-Based Reasoning (CBR) is a problem solving methodology [1]. It is based on similar past problems and solutions to propose a solution to current problem. CBR system is a computer program which simulates human recognition process. Learning capability is an important feature of a CBR system. The accuracy of the system increases with the number of cases in the case library. The paper will illustrate the use of case-based reasoning to assist forecasters to make a better decision.

In Section 5.2, problem formation and data sets for the experiments are described. In Section 5.3, the development of CBR intensity prediction model is given. The system design and architecture are presented in Section 5.4 and the proposed CBR system is evaluated in Section 5.5. Conclusions will follow in the last section.

5.2 Problem formulation

We consider incorporate with related synoptic information to form cases which can be referenced to improve forecast and enable reasoning support. It is particularly useful when satellite data indicate a loosely organized developing system, surface observations of an overall circulation (e.g. Westerly winds on the equatorward side) can help considerably with the satellite interpretation [7].

With reference to [9], we propose an extensive view of forecasting framework as shown in Figure 5.1.

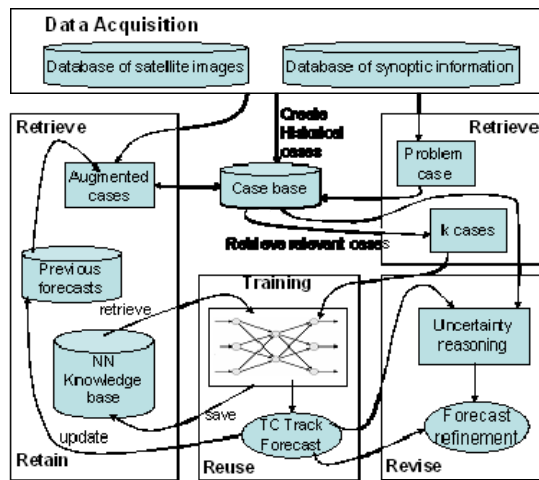


Fig. 5.1. An integrated forecasting framework

A forecaster's subjective judgments regarding the similarity and usefulness of past cases to a current case (case retrieval cycles shown in Fig. 1 will provide explanation as to what, why, when and how particular cases have been selected as best analogs. The use of outranking relations in selecting the best analogs suggests a means for capturing expert knowledge. Past cases are selected based on their degree of match and usefulness to the current situation. This is done by a series of partial matching of past cases with the current case, and by ranking across case dimensions until a smaller set of matching and useful cases are retrieved (e.g. [8]). Past decisions, actions or consequences associated with the best analogs may also be reused or adapted to justify decisions or actions for the current case. The proposed approach therefore enhances learning from experience.

As preliminary study, we have used two sets of data which can be downloaded from the web. One of the datasets is the best track (BT) data from the NOAA National Hurricane Center (NHC) (<http://www.nhc>).

noaa.gov/pastall.shtml). The “best track” means that the data is a comprehensive TC track analysis after considering all available observations and expert interpretation. Another dataset is the optimally interpolated (OI) sea surface temperature (SST) product derived from TRMM/TMI (Tropical Rainfall Measuring Mission/TRMM Microwave Imager) satellite observations. The OI TMI data which could be downloaded at http://www.ssmi.com/sst/microwave_o_i_sst_browse.html is available from January 1998 to present. The files are in binary format consisting of a 0.25 x 0.25 degree (about 25 km) grid (1440 x 720 array) of single byte values representing SST for a given day. The website only provides the access of files in FORTRAN, IDL and MatLab codes respectively. We need to translate the code into JAVA code. Some new fields including ‘IntensityChange’, ‘LandDistance’ and ‘AverageSST’ are created by transforming the raw data in the two datasets. These involve:

1. Filling the intensity change of tropical cyclone
2. Calculating the distance between the TC center and the nearest land mass
3. Calculating the averaged SST within 1° radius of the TC center
4. Discretization on numeric fields in order to uncover the relationship among different fields in the data.

The intensity change of TC is calculated by computing the difference between two consecutive values of tropical cyclone intensity [10]. The new value is stored in the new field called “IntensityChange” in the database:

$$\Delta I_t = I_{t+1} - I_t \quad (5.1)$$

where ΔI_t = TC intensity change at time t ; I_t = TC intensity at time t

The binary file of the OI TMI SST products contains a byte array which has 720 rows and 1440 columns. The center of the first cell of the 1440 column and 720 row map is at 0.125E longitude and -89.875 latitude, and that for the second cell is 0.375E longitude and -89.875 latitude.

The distance between the center of the TC and the nearest land mass is calculated using the land marks (Byte value: 255) in the OI TMI SST products of Remote Sensing System. Great-circle distance is used to calculate the shortest distance between two points on the surface of the earth (Assume the radius of the earth is 6371 km). The value is stored in the new field called “LandDistance” in the database. We compute the sum of the SST within 1° radius of the tropical cyclone center (*) including (X) as:

$$SST(^{\circ}C) = (bytevalue * 0.15) - 3.0 \quad (5.2)$$

The averaged SST is computed by dividing the result by number of (*) including (X). The value is stored in the new field called “AverageSST” in the database. The database contains two tables storing the raw data extracted from the datasets, and transformed data for finding association rules for subsequent experiments.

5.3 Development of CBR intensity prediction model

Apriori algorithm is applied to the data mining process. An ACSII file is generated by Clementine¹ for our case converter to convert all association rules into cases.

5.3.1 Case representation

We represent the case by a 60-bits array. The first 57 bits represent the antecedent of an association rule. The last 3 bits represent the consequent of the association rule. For example, the cumbersome “00000 00000 00100 00000 00010 00000 00000 00000 00000 00000 00000 00100” is used to represent the rule: Wind_4 and AverageSST_High → Intensity-Change_High. If the TC belongs to hurricane category 1 and the averaged sea surface temperature is higher than 28.8 °C, then the TC will intensify in the next 6 hours.

5.3.2 Building a case store

We extract TC data from datasets NHC best track data and OI TMI SST product and have them stored in our TC data base. The system stores cases in a list in the main memory after loading the case file. The case can be retrieved and compared efficiently with a new case because there is no I/O after the loading process. Apriori algorithm is applied to generate association rules and stores in an ASCII file. We extract the consequent and antecedents of each case from the case store.

5.3.3 Case retrieval

To address the case retrieved from the case store, appropriate similarity measure is needed whose purposes are 1) Retrieved cases that can adapt

¹ A datamining toolkit developed by Integral Solutions Ltd: <http://www.isl.co.hk/>

easily to the current problem 2) Retrieved cases that can solve the current problem.

When tropical cyclone arises, we would like to retrieve all cases in the case store that match the weather situation at the time. The matching process can be simple as just checking whether all the case feature values are identical. But this is too raw not suitable for practical applications. We measure the corresponding similarity between TC information including latitude, translation speed, intensity, center pressure, land distance and averaged sea surface temperature for the prediction of change in the next 6 hours.

After the computation of the similarities, we rely on the selection of some threshold (e.g. 0.6 found from experimental testing) to confirm the similar cases. If the similarity is greater than this threshold value, the selected case is identified as one of the best cases. There are three consequents 1) Intensifying Group 2) Stable Group and 3) Weakening Group. Each group has a weight which is the sum of the similarities in the best cases. The consequent of the new case depends on the highest value of three weights.

5.4 Results and discussion

The TCs from 1998 to 2002 in North Atlantic basin are used for data mining. There were more than 200 association rules found by using Apriori algorithm. The minimum antecedent support and minimum rule confidence were 1% and 60% respectively. As shown in Table 5.1, we carry out tests with data for all TCs within the chosen period, TCs within the TC seasons, the exclusion of tropical depression and storm, the use of new binning features (i.e. High, Med, Low) respectively.

Table 5.1. TC Prediction results

	Training Data TCs / Events	Testing Data TCs / Events	Correct Predict.	Predict Acc.
Exp. Result 1	68 / 2202	16 / 550	309	56.18%
Exp. Result 2	68 / 2179	13 / 474	249	52.53%
Exp. Result 3	46 / 1840	11 / 448	259	57.81%
Exp. Result 4	68 / 2202	16 / 550	331	60.18%

The number of correct predictions increases if the actual intensity change of TC is the same with our system's prediction. In result 3, the prediction accuracy is about 58% meaning that our CBR system could predict 58 events correctly out of 100 tropical cyclone events on average.

We note that the association rules for TC which is weakening in the next 6 hours are less than the rules for TC which is intensifying in the next 6 hours. This may be due to the distribution of the data being not normally distributed. More data are biased to being stable and intensifying. The experimental result improves if we filter the tropical depression, tropical storm category A and tropical storm category B. The result of all the TC intensity is less than 64 knots which is too weak and will not be considered. The accuracy also increases if we filter the non-TC seasonal data, or adopt the binning process. Nevertheless, the model appears not doing very well in cases of TC fast intensification, re-intensification and fast weakening. The processing speed of our predicting module is very fast. It depends on the number of cases in the case stores. The time used to predict more than 500 new TC events is less than 1 second including loading the case store under a Pentium 4 class computer with 1GB main memory.

5.5 Conclusion and future work

Tropical cyclone prediction is a very complex process requiring analysis of enormous amount of possibly low quality meteorological data [6]. This study uses CBR to predict the intensity change of the TC. The prediction process is fast because all cases are stored in the main memory. One of the limitations here is that the TC predictor can only predict the intensity change in the next 6 hours. The prediction accuracy of our CBR model may be bound by the data mining process.

For future work, the TC predictor can be extended to provide short-range forecast. More attributes which may be extracted from other datasets can be added to the existing database in order to get better association rules with higher confidences. The accuracy will be improved if the association rules are more reliable [3]. In addition, further work on the fuzziness of those TC parameters may improve the results of prediction (e.g. [4], [8]). Predicting the moving direction of TCs is also important. This is the subject of future paper.

Acknowledgement. The authors would like to acknowledge the partial support of CRG grant G-U186 of The Hong Kong Polytechnic University.

References

1. Aamodt Agnar, Plaza Enric (1994) Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AICom – Artificial Intelligence Communications*, IOS Press, 7(1):39-59
2. Dvorak VF (1975) Tropical cyclone intensity analysis and forecasting from satellite imagery. *Mon. Wea. Rev.* Vol. 103:420-430
3. Feng L, Dillon T, Liu J (2001) Inter-transactional association rules for prediction and their application to studying meteorological data. *Data and Knowledge Engineering*, Elsevier, 37:85-115
4. Hansen B, Riordan D (1998) Fuzzy case-based prediction of ceiling and visibility. In: *Pro. of 1st Conf. on Artificial Intelligence*, American Meteorological Society, pp 118-123
5. Lee RST, Liu JNK (2000) Tropical cyclone identification and tracking system using integrated neural oscillatory elastic graph matching and hybrid RBF network track mining techniques. *IEEE Trans. on NN* 11(3):680-689
6. Li B, Liu J, Dai H (1998) Forecasting from low quality data with applications in weather forecasting. *Int. J. of Comp. & Informatics* 22(3):351-358
7. McGregor JL, Walsh KJ, Katzfey JJ (2000) Climate simulations for Tasmania. In: *Pro. of 4th Int. Conf. on Southern Hemisphere Meteorological and Oceanography*, American Meteorological Society, pp 514-515
8. San Pedro J, Burstein F (2003) A framework for case-based fuzzy multi-criteria decision support for tropical cyclone forecasting. In: *Pro. of 36th Hawaii Int. Conf., System Sciences*
9. Pal Sankar K, Shiu Simon CK (2004) *Foundations of soft case-based reasoning*. Wiley
10. Tang Jiang, Yang Ruixin, Kafatos Menas (2005) Data mining for tropical cyclone intensity prediction. In: *Pro. of 6th Conf. on Coastal Atmospheric and Oceanic Prediction and Processes*, Session 7.5
11. TC Formation Regions, NOAA, http://www.srh.noaa.gov/jetstream/tropics/tc_basins.htm

Part 11

Knowledge engineering, decision rules and data bases

In this part unify papers from the areas of knowledge engineering, decision rules and data bases. One of the major problems in information processing is automated knowledge transfer and utilization, which is a key factor in productivity improvement. Knowledge engineering is a field of science in which the primary goal is to develop techniques and methodologies for achieving automated knowledge transfer and utilization. A decision rule is either a function that maps from the current state to the agent's decision or choice or a mapping from the expressed preferences of each of a group of agents to a group decision. The former is more relevant to decision theory and dynamic optimization whether the later is relevant to game theory. Knowledge engineering and decision rules are major tools for dealing with large databases. Here we present some results about the understanding of knowledge reuse in on-line environment, handling contradictions in knowledge discovery applications, cooperative distributed query-processing approaches, estimating join and projection selectivity factors, connection between the decision rules and metamodel to organize information, a computerized solution for the financial diagnose of the SMEs, tracking deadlocks and phantoms in databases, concurrency control for multilevel secure distributed real-time databases and a new framework for non-deterministic multivalued system minimization.

Chapter 1

Toward an understanding of knowledge reuse in on-line environment

Hsieh-Hua Yang,¹ Jui-Chen Yu,² Hung-Jen Yang³

¹Department of Health Care Administration, Chang Jung Christian University, 396 Chang Jung Rd., Sec. 1, Kway Jen, Tainan, Taiwan, R.O.C., yangsnow@mail.cjcu.edu.tw

²National Science and Technology Museum, #720 Ju-Ru 1st Rd., Kaohsiung, Taiwan, R.O.C., raisin@mail.nstm.gov.tw

³Department of Industrial Technology Education, National Kaohsiung Normal University, #116 Ho-Ping 1st Rd, Kaohsiung City, Taiwan, R.O.C., hjyang@nknucc.nknu.edu.tw

Abstract. For the past ten years, a growing amount of research has shown that the knowledge reuse's role in knowledge management. This paper follows the steps and tries to reveal a model to explain the relation among intention and meta-cognition. In this study, first setup theory model via literature review and an investigating process was conducted by using on-line questionnaire with internal-reliability alpha value of 0.93. In analyzing the collected data, a confirmatory factor analysis was performed on the data using LISREL program. The measurement models were evaluated using maximum likelihood estimation. Convergent and discriminates validity of the items and scales were tested with CFA. Based upon 781 participants' responses, the proposed theory model was proved at well fit conditions. It is concluded that knowledge reuse behavior is affected by both intention and meta-cognitive intention of knowledge reuse.

Keywords. Knowledge Reuse Behavior, CFA, Meta-cognition

1.1 Introduction

Knowledge creation is often viewed as somehow more important, more difficult to manage, and less amenable to information technology support. However, the effective reuse of knowledge is arguably a more frequent organizational concern and one that is clearly related to organizational effectiveness.

Despite its importance, knowledge reuse is something that we know relatively little about. Although knowledge reuse has been observed and researched under many different names in many different settings, findings about knowledge reuse have remained relatively dispersed and un-integrated. One possible explanation is that knowledge reuse is seen as a unitary phenomenon—pretty much the same regardless of who does it, how, and why.

1.1.1 Statement of research problem

Knowledge reuse is a basic component of knowledge management system. While earlier research interest on knowledge management tended to revolve around using information and communication technologies and procedures, attention is increasingly shifted to investigate the role of contextual factors and psychological factors in promoting or inhibiting knowledge reuse. As the importance of knowledge management grows rapidly in most environments, however, few prior or related empirical studies of factors promoting or inhibiting knowledge reuse in on-line environment context exist. The problem of this study was that what are those factors affect individual knowledge reuse propensity within on-learning environment.

1.1.2 Purposes of the study

The purpose of this study was to identify factors affect individual knowledge reuse propensity based upon both contextual factors and individual psychological factors. Specifically, research was designed to answer the following questions:

1. Is individuals' knowledge reuse propensity influenced by their perception of self-interest in an on-line environment?
2. Is individuals' knowledge reuse propensity influenced by their perception of altruism in an on-line environment?

3. Is individuals' knowledge reuse propensity influenced by their perception of self-esteem in an on-line environment?
4. Is individuals' knowledge reuse propensity influenced by their perception of ownership belief in an on-line environment?

1.2 Literature review

1.2.1 Knowledge reuse as a process

The knowledge reuse process can be described in terms of the following stages: capturing or documenting knowledge, packaging knowledge for reuse, distributing or disseminating knowledge (providing people with access to it), and reusing knowledge [1].

Capturing and documenting knowledge can occur in at least four ways.

First, documenting can be a largely passive by-product of the work process, as when virtual teams or communities of practice automatically generate archives of their informal electronic communication that can later be searched.

Second, documenting knowledge for potential reuse can occur within a structure such as that provided by facilitators using brainstorming techniques, perhaps mediated by the use of electronic meeting systems [2].

Third, documenting can involve creating structured records as part of a deliberate, before-the-fact knowledge reuse strategy.

Fourth, documentation can involve a deliberate, after-the-fact strategy of filtering, indexing, packaging, and sanitizing knowledge for later reuse, as in the creation of learning histories, expert help files, or the creation of a data warehouse.

Packaging knowledge is the process of culling, cleaning and polishing, structuring, formatting, or indexing documents against a classification scheme. Among the activities involved in knowledge packaging are: authoring knowledge content, codifying knowledge into "knowledge objects" by adding context [3], developing local knowledge into "boundary objects" by deleting context [4], filtering and pruning content, and developing classification schemes [5].

1.2.2 Meta-cognitive knowledge

The term "meta-cognition" is most often associated with John Flavell [6]. According to Flavell [6, 7], meta-cognition consists of both meta-cognitive

knowledge and meta-cognitive experiences or regulation. Meta-cognitive knowledge refers to acquired knowledge about cognitive processes, knowledge that can be used to control cognitive processes. Flavell further divides meta-cognitive knowledge into three categories: knowledge of person variables, task variables and strategy variables.

Stated very briefly, knowledge of person variables refers to general knowledge about how human beings learn and process information, as well as individual knowledge of one's own learning processes. One may be aware that study session will be more productive if he/she works in the quiet library rather than at home where there are many distractions. Knowledge of task variables includes knowledge about the nature of the task as well as the type of processing demands that it will place upon the individual. For example, you may be aware that it will take more time for you to read and comprehend a science text than it would for you to read and comprehend a novel. Knowledge about strategy variables include knowledge about both cognitive and meta-cognitive strategies, as well as conditional knowledge about when and where it is appropriate to use such strategies.

1.2.3 Meta-cognitive regulation

Meta-cognitive experiences involve the use of meta-cognitive strategies or meta-cognitive regulation [8]. Meta-cognitive strategies are sequential processes that one uses to control cognitive activities, and to ensure that a cognitive goal (e.g., understanding an on-line text) has been met. These processes help to regulate and oversee learning, and consist of planning and monitoring cognitive activities, as well as checking the outcomes of those activities.

For example, after reading a paragraph in an on-line platform a learner may question herself about the concepts discussed in the paragraph. Her cognitive goal is to understand the text. Self-questioning is a common meta-cognitive comprehension monitoring strategy. If she finds that she cannot answer her own questions, or that she does not understand the material discussed, she must then determine what needs to be done to ensure that she meets the cognitive goal of understanding the text. She may decide to go back and re-read the paragraph with the goal of being able to answer the questions she had generated. If, after re-reading through the text she can now answer the questions, she may determine that she understands the material. Thus, the meta-cognitive strategy of self-questioning is used to ensure that the cognitive goal of comprehension is met. Information technology may play assistance role to help learners with searching, indexing, storing, or marking.

1.2.4 Meta-cognitive strategies

Most definitions of meta-cognition include both knowledge and strategy components; however, there are a number of problems associated with using such definitions. One major issue involves separating what is cognitive from what is meta-cognitive.

Flavell himself acknowledges that meta-cognitive knowledge may not be different from cognitive knowledge [6]. The distinction lies in how the information is used.

Recall that meta-cognition is referred to as "thinking about thinking" and involves overseeing whether a cognitive goal has been met. This should be the defining criterion for determining what is meta-cognitive. Cognitive strategies are used to help an individual achieve a particular goal while meta-cognitive strategies are used to ensure that the goal has been reached. Meta-cognitive experiences usually precede or follow a cognitive activity. They often occur when cognitions fail, such as the recognition that one did not understand what one just read. Such an impasse is believed to activate meta-cognitive processes as the learner attempts to rectify the situation [9].

Meta-cognitive and cognitive strategies may overlap in that the same strategy, such as questioning, could be regarded as either a cognitive or a meta-cognitive strategy depending on what the purpose for using that strategy may be. For example, you may use a self-questioning strategy while reading as a means of obtaining knowledge (cognitive), or as a way of monitoring what you have read (meta-cognitive). Because cognitive and meta-cognitive strategies are closely intertwined and dependent upon each other, any attempt to examine one without acknowledging the other would not provide an adequate picture.

Knowledge is considered to be meta-cognitive if it is actively used in a strategic manner to ensure that a goal is met.

Meta-cognition or the ability to control one's cognitive processes (self-regulation) has been linked to knowledge reuse. By applying information technology in an on-line environment, knowledge reuse behavior would be affected. Meta-cognition is executive processes that control other cognitive components as well as receive feedback from these components. It is responsible for figuring out how to do a particular task or set of tasks, and then making sure that task or set of tasks are done correctly. These executive processes involve knowledge reuse behavior.

Meta-cognition enables learners to benefit from instruction and influences the use and maintenance of cognitive strategies. Meta-cognitive Strategies for Successful Knowledge Learning are:

Consciousness:

- Consciously identify what you already know
- Define the learning goal
- Consider your personal resources (e.g. textbooks, access to the library, access to a computer work station or a quiet study area)
- Consider the task requirements (essay test, multiple choice, etc.)
- Determine how your performance will be evaluated
- Consider your motivation level
- Determine your level of anxiety

Planning:

- Estimate the time required to complete the task
- Plan study time into your schedule and set priorities
- Make a checklist of what needs to happen when
- Organize materials
- Take the necessary steps to learn by using strategies like outlining, mnemonics, diagramming, etc.

Monitoring and Reflection:

- Reflect on the learning process, keeping track of what works and what doesn't work for you
- Monitor your own learning by questioning and self-testing
- Provide your own feedback
- Keep concentration and motivation high

As knowledge workers become more skilled at using meta-cognitive strategies, they gain confidence and become more independent as learners. Independence leads to ownership as worker's realize they can pursue their own intellectual needs and discover a world of information at their fingertips. The task of technology for knowledge management is to acknowledge, cultivate, exploit and enhance the meta-cognitive capabilities of all knowledge workers.

1.3 Theory frame

Based upon the theory of information reuse [10, 11] and meta-cognition theory [12, 13], a multilevel model was constructed to simultaneously investigate these factors which in both theories could impact individuals' knowledge reuse intentions.

Information reuse theory identified social and individual antecedents of knowledge reuse, and at the same time it distinguishes information product from expertise to explain that people reuse knowledge has a different motivation.

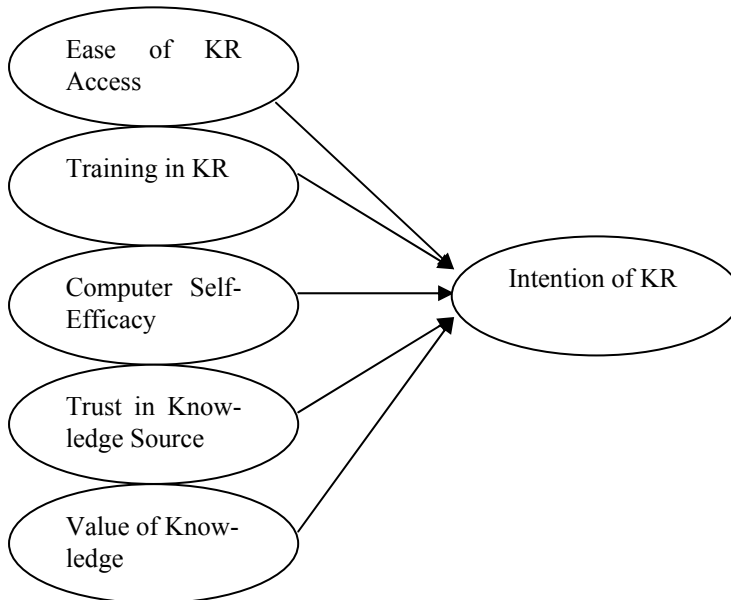


Fig. 1.1. Theory model of intention of knowledge reuse

Meta-cognition theory helps us to understand that knowledge reuse is conducted according to learner's evaluation via strategy, reflection, and self-regulation.

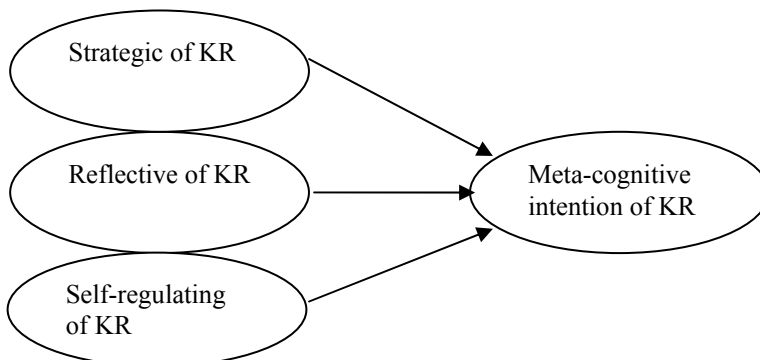


Fig. 1.2. Theory model of meta-cognitive intention of knowledge reuse

1.3.1 Intention of knowledge reuse

Intention of knowledge reuse relate to personal perception of easy use. According to technology acceptance model, TAM, easy of use is a factor contributes to technology attitude [14]. Under TAM, easy of knowledge access can be seen as an indicator of ease of use, ant thus, as a predictor of the user's intention to use and ultimate use of the knowledge system.

Hypothesis 1: Easy of knowledge access will be positively related to the intention of knowledge reuse.

One way the knowledge workers could raise their expectancy perceptions is through adequate training in the activities they are expected to perform. With advanced training, knowledge workers are more likely to achieve the outcome, or get better results, with the same level of effort.

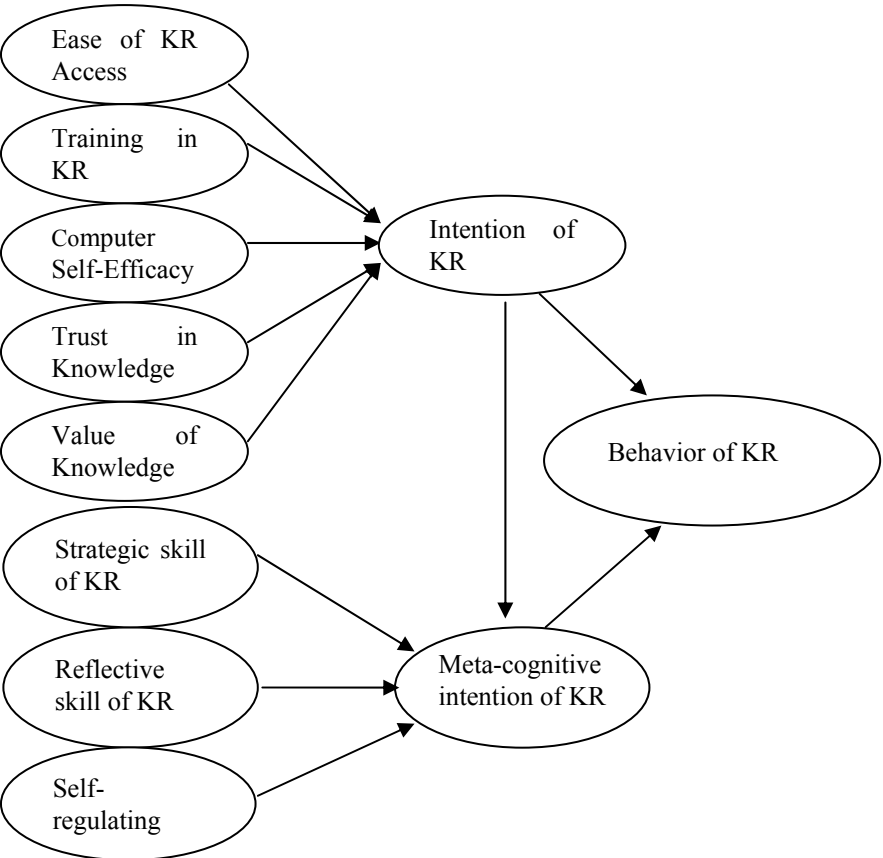


Fig. 1.3. Theory model of knowledge reuse behavior

If people are asked to engage in activities they are not capable of performing, their expectancy perceptions will be low and they will be less likely to perform the action [15].

Training in knowledge reuse is a key means of informing knowledge workers about the sources of existing knowledge available to them and how to go about accessing them. Thus, increased training in knowledge reuse should result in a higher level of motivation to reuse knowledge.

Hypothesis 2: Training in knowledge reuse will be positively related to the intention of knowledge reuse.

Use of knowledge management systems requires competency in computer usage. Computer self-efficacy refers to individuals' beliefs about their abilities to competently use computers [16]. The more confident one is in his or her ability to use information technology, the greater the likelihood he or she will believe that the effort involved in accessing computerized knowledge databases will result in a successful outcome, which here would be accessing relevant knowledge.

Hypothesis 3: Computer self-efficacy will be positively related to the frequency of knowledge reuse.

In the context of knowledge reuse, instrumentality is the belief that one will get relevant, reusable knowledge from successfully accessing the existing knowledge base. In this study, instrumentality essentially refers to the belief that access to the knowledge will result in obtaining useful knowledge. The more an individual trusts that the knowledge in the database of existing knowledge is timely and accurate, the greater the belief that accessing the database will result in the reward of a useful piece of knowledge. Trust in the system can be viewed as reflecting the perceived usefulness of that system, and therefore, under the TAM, trust would be positively associated with the use of the system. Thus, there should be a positive relationship between trust in the knowledge and the intention of knowledge reuse.

Hypothesis 4: Trust in knowledge source will be positively related to the intention of knowledge reuse.

Whereas instrumentality refers to the belief that one will get something for achieving the outcome, another key aspect should be considered is the values of what one will receive for achieving the outcome. The greater the valence is, the higher the motivation for engaging in the behavior will be. Therefore, we expect there to be a positive relationship between the per-

ceived value of the knowledge received through knowledge reuse and the intention of knowledge reuse.

Hypothesis 5: Value of knowledge will be positively related to the intention of knowledge reuse.

1.3.2 Meta-cognitive intention of knowledge reuse

Knowing how to learn, and knowing which strategies work best, are valuable skills that differentiate expert learners from novice learners. Meta-cognition is a critical ingredient to successful learning knowledge. It consists of two basic processes occurring simultaneously: monitoring your progress as you learn, and making changes and adapting your strategies if you perceive you are not doing so well.

Meta-cognitive skills include taking conscious control of learning, planning and selecting strategies, monitoring the progress of learning, correcting errors, analyzing the effectiveness of learning strategies, and changing learning behaviors and strategies when necessary[17]. Strategic, reflective, and self-regulating are three components of meta-cognitive skills. These skills become factors affect the meta-cognitive intention of knowledge reuse. Thus, there exists relationship between strategic and meta-cognitive intention of knowledge reuse.

Hypothesis 6: Strategic skill will be positively related to the intention of meta-cognitive intention of knowledge reuse.

Hypothesis 7: Reflective skill will be positively related to the intention of meta-cognitive intention of knowledge reuse.

Hypothesis 8: Self-regulating skill will be positively related to the intention of meta-cognitive intention of knowledge reuse.

1.3.3 Behavior of knowledge reuse

The behavior of knowledge reuse is to conduct capturing and packaging knowledge for reuse [1]. The behavior is affected by the intention of knowledge reuse and meta-cognitive.

The frequency of knowledge behavior would be positively affected by one's intention of reusing knowledge. Whenever a person intended to reuse certain knowledge, he/she would re-think what is pursued. At this moment, the meta-cognition process would begin. Thus, it is proposed that

reuse intention affects reuse behavior. The reuse intention also affects meta-cognitive intention. The behavior is also affected by the intention of meta-cognition.

Hypothesis 9: Knowledge Reuse Intention will be positively related to the reuse behavior.

Hypothesis 10: Knowledge Reuse Intention will be positively related to the meta-cognitive intention.

Hypothesis 11: The meta-cognitive intention of knowledge reuse will be positively related the reuse behavior.

1.4 Research method

1.4.1 Data collection

Data to test the model and hypotheses were drawn from a cross-sectional field study of students in National Kaohsiung Normal University in Taiwan. The knowledge reuse behavior data were acquired from online-learning platform that students used for academic learning activity. There were total 781 subjects in this study.

1.4.2 Measurement of constructs

Constructs were measured by using multiple-item scales; a pilot study of 80 subjects was conducted to ensure reliability of questionnaire. The overall internal consistence value is 0.93 and the sub-domain value is from 0.78 to 0.85. There were twenty two items in the questionnaire. All items use seven—point Likert scales anchored between “strongly disagree” and “strongly agree”. There were three levels of factors. The detail structure was listed in Table 1.1.

1.4.3 Scale validation method

In analyzing the collected data, a confirmatory factor analysis was performed on the data using LISREL program. The measurement models were evaluated using maximum likelihood estimation. Convergent and discriminates validity of the items and scales were tested with CFA.

Table 1.1. Research variables lists according to factor and level

Levels	Factors	Variables
1 Intention of KR	Ease of KR Access	EA1
		EA2
	Training in KR	Train1
		Train2
	Computer Self-Efficacy	CSE1
		CSE2
	Trust in Knowledge	Tru1
		Tru2
	Value of Knowledge	Value1
		Value2
1 Meta-cognitive Intention of KR	Strategic of KR	ST1
		ST2
	Reflective of KR	Ref1
		Ref2
	Self-regulating of KR	Reg1
		Reg2
2	Intention of KR	Inten1
		Inten2
		Inten3
2	Meta-cognitive intention of KR	Meta1
		Meta2
		Meta3
3	Behavior of KR	BKR1
		BKR2
		BKR3

1.5 Data analysis and results

1.5.1 Samples

There were 781 participants completed the survey via online website. There were 438 female and 343 male. The percentage was listed in Table 1.2.

Table 1.2. Gender distribution of participants

		Frequency	Percent	Valid Percent
Valid	Female	438	56.1	56.1
	Male	343	43.9	43.9
	Total	781	100.0	100.0

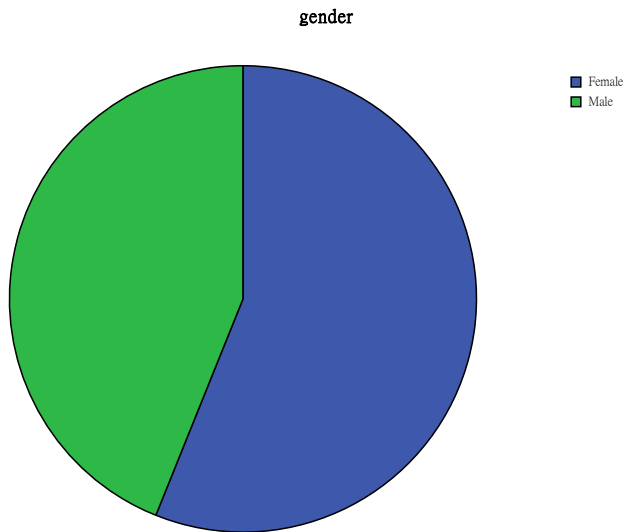


Fig. 1.4. Gender information of respondents

1.5.2 Measurement equations

The LISREL estimates results were listed in Tables 1.3 – 1.7. In Table 1.3, the factor of intent1 would significantly contribute to intention of knowledge reuse and predict at forty-one percent.

Table 1.3. Measurement equations of Intention of Knowledge Reuse

inten1 = 0.65*IKR, Errorvar.= 0.61 , R ² = 0.41		
	(0.025)	
	24.55	
inten2 = 0.91*IKR, Errorvar.= 0.27 , R ² = 0.76		
(0.032)	(0.019)	
28.63	14.37	
inten3 = 0.52*IKR, Errorvar.= 0.92 , R ² = 0.23		
(0.022)	(0.051)	
23.46	18.00	

The factor of inten2 would significantly contribute to intention of knowledge reuse and predict at seventy-six percent. The factor of intent3 would significantly contribute to intention of knowledge reuse and predict at twenty-three percent.

In Table 1.4, the factor of meta1 would significantly contribute to intention of knowledge reuse and predict at twenty-six percent.

Table 1.4. Measurement equations of Meta-cognitive Intention of Knowledge Reuse

meta1 = 0.55*Metacog, Errorvar.= 0.83 , R ² = 0.26
(0.034)
24.11
meta2 = 0.91*Metacog, Errorvar.= 0.21 , R ² = 0.80
(0.031) (0.016)
29.37 13.19
meta3 = 0.95*Metacog, Errorvar.= 0.12 , R ² = 0.89
(0.031) (0.012)
30.27 9.44

The factor of meta2 would significantly contribute to intention of knowledge reuse and predict at eighty percent. The factor of meta3 would significantly contribute to intention of knowledge reuse and predict at eighty-nine percent.

In Table 1.5, the factor of bkr1 would significantly contribute to intention of knowledge reuse and predict at fifty-three percent. The factor of bkr2 would significantly contribute to intention of knowledge reuse and predict at eighty-five percent. The factor of bkr3 would significantly contribute to intention of knowledge reuse and predict at eighty-four percent.

Table 1.5. Measurement equation of Behavior of Knowledge Reuse

bkr1 = 0.79*BKR, Errorvar.= 0.55 , R ² = 0.53
(0.026)
20.98
bkr2 = 0.94*BKR, Errorvar.= 0.15 , R ² = 0.85
(0.023) (0.016)
41.55 9.58
bkr3 = 0.93*BKR, Errorvar.= 0.16 , R ² = 0.84
(0.013) (0.012)
72.10 13.95

In Table 1.6, the first level factors on measuring knowledge reuse intention were listed and the contribution of factors is between thirty-eight and eighty-one percent.

In Table 1.7, the first level factors on measuring meta-cognitive knowledge reuse intention were listed and the contribution of factors is between twenty-three and eighty-six percent.

All factors in the models were significant contribute in predicting. These findings demonstrate the relationships in these models exist.

Table 1.6. Measurement equations the first level factors of Intention of Knowledge Reuse

ea1 = 0.66*EaseAcce, Errorvar.= 0.59 , R ² = 0.42	
(0.032)	(0.031)
20.41	18.99
ea2 = 0.88*EaseAcce, Errorvar.= 0.32 , R ² = 0.71	
(0.030)	(0.025)
29.09	12.86
train1 = 0.65*Training, Errorvar.= 0.67 , R ² = 0.38	
(0.027)	(0.023)
23.63	29.76
train2 = 0.84*Training, Errorvar.= 0.34 , R ² = 0.67	
(0.032)	(0.023)
26.36	14.93
cse1 = 1.05*CSE, Errorvar.= 0.26 , R ² = 0.81	
(0.034)	(0.037)
30.46	6.93
cse2 = 0.82*CSE, Errorvar.= 0.44 , R ² = 0.60	
(0.033)	(0.031)
24.64	14.04
tru1 = 0.96*Trust, Errorvar.= 0.26 , R ² = 0.78	
(0.034)	(0.035)
28.37	7.56
tru2 = 0.97*Trust, Errorvar.= 0.27 , R ² = 0.78	
(0.034)	(0.034)
28.38	7.88
value1 = 0.99*Value, Errorvar.= 0.37 , R ² = 0.73	
	(0.030)
	12.43
value2 = 0.78*Value, Errorvar.= 0.29 , R ² = 0.68	
(0.031)	(0.020)
24.69	14.00

Table 1.7. Measurement equations the first level factors of Meta-cognitive Intention of Knowledge Reuse

st1 = 1.05*Strategi, Errorvar.= 0.35 , R ² = 0.76	
(0.026)	(0.020)
40.83	17.51
st2 = 0.63*Strategi, Errorvar.= 0.71 , R ² = 0.36	
(0.032)	(0.037)
19.98	19.40
ref1 = 1.09*Reflecti, Errorvar.= 0.20 , R ² = 0.86	
(0.025)	(0.022)
43.12	8.93

ref2 = 0.41*Reflecti, Errorvar.= 0.51 , R ² = 0.25	
(0.026)	(0.027)
15.51	18.84
reg1 = 0.51*Self_reg, Errorvar.= 0.89 , R ² = 0.23	
(0.023)	(0.026)
22.53	34.00
reg2 = 0.66*Self_reg, Errorvar.= 0.63 , R ² = 0.41	
(0.020)	(0.028)
33.13	22.65

1.5.3 Structural equations

In Table 1.8, the structural equations of knowledge reuse behavior are listed. The contributions of equations are sixty-nine, ninety-seven, and eighty-three percent.

Table 1.8. Structural equations of knowledge reuse behavior

IKR = 0.039*Value + 3.52*EaseAcce - 2.82*Training + 0.25*CSE + 0.41*Trust, Errorvar.= 0.31, R ² = 0.69
Metacog = 0.60*IKR + 0.68*Self_reg - 3.75*Reflecti + 3.00*Strategi, Errorvar.= 0.027, R ² = 0.97
BKR = - 0.031*IKR + 0.95*Metacog, Errorvar.= 0.15, R ² = 0.85
Value = 0.91*Strategi, Errorvar.= 0.17 , R ² = 0.83

In Table 1.9, the reduced form equations of knowledge reuse behavior model were listed. For model IKR, there are five factors. For model meta-cognitive, there are seven factors. For model behavior, there also are seven factors. For model value, there is one factor.

Table 1.9. The reduced form equations of knowledge reuse behavior model

IKR= 3.52*EaseAcce - 2.82*Training + 0.25*CSE + 0.41*Trust + 0.036*Strategi, Errorvar.= 0.32, R ² = 0.68
Metacog = 2.12*EaseAcce + 0.68*Self_reg - 3.75*Reflecti - 1.70*Training + 0.15*CSE + 0.24*Trust + 3.02*Strateg, Errorvar.= 0.14, R ² = 0.86
BKR = 1.90*EaseAcce + 0.65*Self_reg - 3.56*Reflecti - 1.53*Training + 0.13*CSE + 0.22*Trust + 2.86*Strategi, Errorvar.= 0.27, R ² = 0.73
Value = 0.91*Strategi, Errorvar.= 0.17, R ² = 0.83

1.5.4 Conclusion

Based on the LIRREL estimate measurement equations, it is concluded that all eleven hypothesis were hold truly. These prove all separate rela-

tions are existed. According to all three structural equations, theory model as a whole is also existed.

Based upon these findings, Knowledge reuse behavior is affected by both the intention and meta-cognitive intention of knowledge reuse and the meta-cognitive intention is affected by intention of knowledge reuse.

References

1. Alavi M (2000) Managing organizational knowledge. OH: Pinnaflex Educational Resources
2. Dean DL, Lee JD, Pendergast MO, Hickey AM, Nunamaker JF (1997-1998) Enabling the effective involvement of multiple users: methods and tools for collaborative software engineering. *Journal of Management Information Systems* 14(3):179-222
3. Roth G, Kleiner A (1998) Developing organizational memory through learning histories. *Organizational Dynamics* 27:43-59
4. Andreoli JM, Borghoff UM, Pareschi R (1996) The Constraint-Based Knowledge Broker Model - Semantics, Implementation and Analysis. *Journal of Symbolic Computation* 21:635-667
5. Manville B (1999) A complex adaptive approach to KM: reflections on the ease of McKinsey & Company, Inc.. *Knowledge Management Review*, pp 26-31
6. Flavell JH (1979) Metacognition and cognitive monitoring: A new area of cognitive-developmental inquiry. *American Psychologist* 34
7. Flavell JH (1987) Speculations about the nature and development of metacognition. Hillsdale, Lawrence Erlbaum Associates, New Jersey
8. Brown AL (1987) Metacognition, executive control, self-regulation, and other more mysterious mechanisms. In: Weinert FE, Kluwe RH (eds) *Metacognition, motivation, and understanding*. Hillsdale, Lawrence Erlbaum Associates, New Jersey, pp 65-116
9. Roberts MJ, Erdos G (1993) Strategy selection and metacognition. *Educational Psychology* 13:259-266
10. Watson S, Hewett K (2006) A multi-theoretical model of knowledge transfer in organizations: Determinants of knowledge contribution and knowledge reuse. *Journal of Management Studies* 43:141-173
11. Lee HJ, Ahn HJ, Kim JW, Park SJ (2006) Capturing and reusing knowledge in engineering change management: A case of automobile development. *Information Systems Frontiers* 8:375-394
12. Call J (2003) On linking comparative metacognition and theory of mind. *Behavioral & Brain Sciences* 26:341-342
13. Borkowski JG (1996) Metacognition - Theory or chapter heading. *Learning & Individual Differences* 8:391-402

14. Davis FD, Bagozzi RP, Warshaw PR (1989) User acceptance of computer technology: a comparison of two theoretical models. *Management Science* 35:982-1033
15. Pinder CC (1997) *Work Motivation in Organizational Behavior*. Upper Saddle River, NJ, Prentice-Hall
16. Compeau DR, Higgins CA (1995) Computer self-efficacy: development of a measure and initial test. *MIS Quarterly* 19:189-211
17. Ridley DS, Schutz PA, Glanz RS, Weinstein CE (1992) Self-regulated learning: the interactive influence of metacognitive awareness and goal-setting. *Journal of Experimental Education* 60:293-306

Chapter 2

Handling contradictions in knowledge discovery applications

David Anderson

School of Computing, University of Portsmouth, Portsmouth PO1 2EG,
Ph. (+44) 2392 84 6668, david.anderson@port.ac.uk

Abstract. There have been a number of very credible attempts to devise paraconsistent logics to deal with the problems caused by the unavoidability of contradictions in knowledge bases and elsewhere. This paper suggests a set of principles for generating logical operator semantics which are broadly drawn from classical logic. These lead directly to a paraconsistent logic, LM_4 , which significantly outperforms other systems and solves the problem for all practical purposes without giving rise to the difficulties inherent in other attempted solutions.

Keywords. Paraconsistency, logic, AI, reasoning, contradiction-handling, theorem-proving, machine-deduction, semantic tableaux

2.1 What's wrong with contradictions?

Let $\vdash$ be a relation of logical consequence. We may say that $\vdash$ is *explosive* iff for arbitrary formulae q and r , $(q \ \& \ \neg q) \vdash r$. Classical logic, intuitionistic logic, and most other standard logics are explosive but ordinary human reasoning is not. The explosive character of standard logic has long been a cause of concern among mathematicians and logicians whose early attempts to address the difficulty are summed up by Quine thus:

“Consider... the crisis which was precipitated in the foundations of mathematics, at the turn of the century, by the discovery of Russell's paradox

and the other antinomies of set theory. These contradictions had to be obviated by ad hoc devices; our mathematical myth-making had become deliberate and evident to all." [1]

It is in very much the same spirit of dissatisfaction with attempts to avoid rather than to solve the problem which inconsistency poses, that Graham Priest attempts to incorporate, in a controlled way, paradoxes into the very heart of a system of reasoning by formulating a logic (LP) in which the relation of logical consequence is not explosive. Logics displaying this feature are said to be paraconsistent and there are quite a few variants available.

2.2 Correcting classical logic's mistake?

According to Priest, classical logic is founded, at least in part on an error:

"Classical logic errs in assuming that no sentence can be both true and false. We wish to correct this assumption. If a sentence is both true and false, let us call it 'paradoxical' (p). If it is true but not false we will call it 'true only' (t) and similarly for 'false only' (f)." [2]

We are further told that:

"A sentence is true iff its negation is false, hence the negation of a true and false sentence is false and true, i.e., paradoxical. The negation of a true only sentence is false only ... the negation of a false only sentence is true only." [2]

On the basis of these assertions and more in the same vein Priest presents us with the following tables for negation, conjunction and disjunction:

Table 2.1. Negation and Conjunction tables in LP

$\neg$		$\&$	t	p	f
t	f	t	t	p	f
p	p	p	p	p	f
f	t	f	f	p	f

Table 2.2. Disjunction and Material Implication tables in LP

V	t	p	f	$\Rightarrow$	t	p	f
t	t	t	t	t	t	p	f
p	t	p	p	p	t	p	p
f	t	p	f	f	t	t	t

In LP both t and p are designated values because both are the values of 'true' sentences. The intended effect of these innovations is to render invalid *ex contradictione quodlibet* $[q \ \& \ \neg q \mid - \ r]$ but it is not possible to remove single theorems from logic and unfortunately for LP a number of valuable forms of reasoning get lost in the process. Other significant casualties include:

$q, \neg q \vee r$	$\mid -$	r	<i>Disjunctive Syllogism</i>
$q \Rightarrow (r \ \& \ \neg r)$	$\mid -$	$\neg q$	<i>Reductio Ad Absurdum</i>
$q \Rightarrow r, r \Rightarrow s$	$\mid -$	$q \Rightarrow s$	<i>Transitivity</i>
$q, q \Rightarrow r$	$\mid -$	r	<i>Modus Ponens</i>
$q \Rightarrow r, \neg r$	$\mid -$	$\neg q$	<i>Modus Tollens</i>

It does not require a great deal of specialised interest in formal logic to realise that the loss of so many common forms of reasoning represents a significant inhibition on the power and attractiveness of the resultant system. Indeed, Priest admits that "We cannot give these inferences up without crippling classical reasoning" [4] but putting a brave face on matters suggests that we should regard the inference forms above not as invalid (which in reality they are) but as 'quasi-valid'. 'Quasi' because they are truth-preserving provided that there are no paradoxical sentences involved. We are encouraged to accept the following methodological maxim:

"MM

Unless we have specific grounds for believing that paradoxical sentences are occurring in our argument, we can allow ourselves to use both valid and quasi-valid inferences." [4]

The methodological maxim is an admission that LP cannot realistically be used as the logic of first choice nor, consequently, as the basis of an inference engine for normal use by an autonomous machine.

Discomfort with endorsing a rejection of the principle of bivalence on which LP, and many other multi-valued systems, depends may go some way to explaining why the development of paraconsistent logics as a response to the problem of inconsistency has been less than universally embraced. The idea of a sentence being literally 'true and false' strikes at the heart of the very notions of truth and falsity and it would be a rash logician indeed who did not hope for an alternative approach. Perhaps if the deductive power of LP were greater it would be enough to sweeten the pill but as things stand the temptation is hardly overwhelming.

2.3 A different approach - knowledge bases and presence values

Perhaps we would profit from approaching this problem from a slightly different angle. Anyone who has recently considered seriously building an autonomous intelligent machine will have turned their minds to the problems caused by the inevitability of inconsistency getting into the machine's knowledge base. One approach following David Lewis [4] might be to keep a multitude of small knowledge bases each relatively independent from the others in hopes that when contradiction does arise it might have only limited impact and when discovered might be relatively easy to correct. This, in essence, is the approach Lenat and Guha adopt in CYC:

"There is no need – and probably not even any possibility - of achieving a global consistent unification of... [a] very large KB... How should the system cope with inconsistency? View the knowledge space, and hence the KB, not as one rigid body, but rather as a set of independently supported buttes... [As] with large office buildings, independent supports should make it easier for the whole structure to weather tremors such as local anomalies ... [I]nferring an inconsistency is only slightly more serious than the usual sort of 'dead end' a searcher runs into..." [5]

Copeland [6] is surely right to be uneasy about such a cavalier approach, for if the information contained in the buttes is available to other buttes (as is the case with CYC) then the effect of contradiction will spread. However if the buttes are completely independent then it is difficult to see how they can in any real sense act as part of a larger whole.

All is not lost. Whatever reservations logicians might have about saying that a sentence can literally be true and false there can be no such objections raised to saying that a given knowledge base K contains both a proposition q and its negation $\neg q$. After all, this is just another way of saying that that K contains a contradiction. It is perfectly possible in the face of this information about K to say q is actually true (or false) and that $\neg q$ (or q) ideally should be removed from K . Of course, we could follow Priest and say q is paradoxical and K merely reflects this fact or we need not actually make any truth claims at all. Certainly if K contains q but does not contain $\neg q$, any system relying solely on K would assert q if interrogated, but that does not settle the truth value of q . As system builders we need not take sides concerning the truth value of the potential constituent elements of knowledge bases although we do have an interest in what may be called the *presence* values of those elements.

Thus, let L be a propositional language whose set of propositional variables is P .

For any given propositional variable q of P , let $M(q)$ be a mapping of q onto $\{in, out\}$.

A knowledge base K , is specified by the mapping of each of the propositional variables comprising P onto $\{in, out\}$. Suppose L has just three propositional variables q, r, s and

$M(q) \Rightarrow \{in\}$, $M(r) \Rightarrow \{in\}$, and $M(s) \Rightarrow \{out\}$ then K will be $\{q, r\}$

That is to say, for any proposition q , which may be expressed in our propositional language then either q is present in a given knowledge base K or it is not. We may think of this as the principle of presence-bivalence.

Reasoning, from the view of programs using knowledge bases, can be thought of as a matter of expanding (and contracting¹) knowledge bases. To regulate this process let us permit the expansion of the knowledge base in accordance with the following seven principles:

- Negation
 1. Any knowledge base for which $M(q) = \{in\}$ may be extended so that $M(\neg q) = \{in\}$ and vice versa.
- Conjunction
 2. Any knowledge base for which $M(q) = \{in\}$ and $M(r) = \{in\}$ may be extended so as $M(q \& r) = \{in\}$
 3. Any knowledge base for which $M(q) = \{out\}$ or $M(r) = \{out\}$ may be extended so as $M(\neg(q \& r)) = \{in\}$
- Disjunction
 4. Any knowledge base for which $M(q) = \{in\}$ or $M(r) = \{in\}$ may be extended so as $M(q \vee r) = \{in\}$
 5. Any knowledge base for which $M(q) = \{out\}$ and $M(r) = \{out\}$ may be extended so as $M(\neg(q \vee r)) = \{in\}$
- Material implication
 6. Any knowledge base for which $M(q) = \{out\}$ and $M(\neg q) = \{in\}$ or for which $M(r) = \{in\}$ and $M(\neg r) = \{out\}$ may be extended so as $M(q \Rightarrow r) = \{in\}$

¹ I will not deal here with the issue of contracting knowledge bases as it has little bearing on the main issue I want to address.

7. Any knowledge base for which $[M(r) = \{\text{out}\} \text{ or } M(\neg r) = \{\text{in}\}]$ and $\neg[M(q) = \{\text{out}\} \text{ and } M(\neg q) = \{\text{in}\}]$ may be extended so as $M(\neg(q \Rightarrow r)) = \{\text{in}\}$

Now, let $v(q) \Rightarrow \{a, d\}$ (i.e., v is a presence valuation of the propositional variables) according to the following rules

- R1. If $M(q) \Rightarrow \{\text{in}\}$ and $M(\neg q) \Rightarrow \{\text{out}\}$ then $v(q) = a$ (assert)
 R2. If $M(q) \Rightarrow \{\text{out}\}$ and $M(\neg q) \Rightarrow \{\text{in}\}$ then $v(q) = d$ (deny)

Applying the seven principles of expansion and the valuation rules yields the following presence-tables:

Table 2.3. Two-valued presence tables for Negation and Conjunction

$\neg$		$\&$	a	d
a	d	a	a	d
d	a	d	d	d

Table 2.4. Two-valued presence tables for Disjunction and Material Implication

V	a	d	$\Rightarrow$	a	d
a	a	a	a	a	d
d	a	d	d	a	a

Let v^+ be the natural extension of v to all the sentences of L using the tables above.

If E is a set of sentences of L , we define;

- $E \models q$ iff there is no $v(q) = d$, but for all $r \in E$, $v^+(r) = a$
 $\models q$ iff for all v , $v^+(q) \not\leq d$

Note that if one wants to read assert as “true” and deny as “false” these are the familiar tables of classical logic and if ‘a’ is designated as above we have created a perfectly standard classical system.

So far, so good. However, R1 and R2 do not cater for all the situations which might arise. Specifically we have not considered the following situations:

- Case 3: $M(q) = \{\text{out}\}$ and $M(\neg q) = \{\text{out}\}$
 Case 4: $M(q) = \{\text{in}\}$ and $M(\neg q) = \{\text{in}\}$.

If K contains q and also contains $\neg q$ (Case 4) a system relying solely on K would equivocate with respect to q . If K does not contain q and also does not contain $\neg q$ (Case 3) a system relying solely on K may be said to be indifferent with respect to q .

Thus, let $v(q) \Rightarrow \{a, d, e, i\}$ (i.e., v is a presence valuation of the propositional variables) according to the following rules

- R1. If $M(q) \Rightarrow \{in\}$ and $M(\neg q) \Rightarrow \{out\}$ then $v(q) = a$ (assert)
- R2. If $M(q) \Rightarrow \{out\}$ and $M(\neg q) \Rightarrow \{in\}$ then $v(q) = d$ (deny)
- R3. If $M(q) \Rightarrow \{in\}$ and $M(\neg q) \Rightarrow \{in\}$ then $v(q) = e$ (equivocate)
- R4. If $M(q) \Rightarrow \{out\}$ and $M(\neg q) \Rightarrow \{out\}$ then $v(q) = i$ (indifferent)

This yields the following tables:

Table 2.5. Four-valued presence tables for Negation and Conjunction

$\neg$		$\&$	a	d	e	i
a	d	a	a	d	e	i
d	a	d	d	d	d	d
e	e	e	e	d	e	d
i	i	i	i	d	d	i

Table 2.6. Four-valued presence tables for Disjunction and Material Implication

V	a	d	e	i	$\Rightarrow$	a	d	e	i
a	a	a	a	a	a	a	d	d	d
d	a	d	e	i	d	a	a	a	a
e	a	e	e	a	e	a	d	d	d
i	a	i	a	i	i	a	d	d	d

In this case let us designate both a and e , so let v^+ be the natural extension of v to all the sentences of L using the tables above.

If E is a set of sentences of L , we define;

$E \vdash q$ iff there is no $v(q) = d$, but for all $r \in E$, $v^+(r) = a$ or e
 $\vdash q$ iff for all v , $v^+(q) \not\leq d$

The features of this system make it a very powerful tool. Like LP, LM_4 is not explosive as *Ex Contradictione Quodlibet* $[(q \& \neg q) \vdash r]$ is LM_4 invalid as is *Disjunctive Syllogism* $[q, \neg q \vee r \vdash r]$ but the following familiar forms are preserved (a longer list of results is included as an appendix):

$q \Rightarrow (r \& \neg r)$	$\vdash$	$\neg q$	<i>Reductio Ad Absurdum</i>
$q \Rightarrow r, r \Rightarrow s$	$\vdash$	$q \Rightarrow s$	<i>Transitivity</i>
$q, q \Rightarrow r$	$\vdash$	r	<i>Modus Ponens</i>
$q \Rightarrow r, \neg r$	$\vdash$	$\neg q$	<i>Modus Tollens</i>
$q \& r$	$\vdash$	$q \mid r$	<i>And Elimination</i>
q, r	$\vdash$	$q \& r$	<i>And Introduction</i>

q	$\vdash$	$q \vee r$	<i>Or Introduction</i>
$q \vee r, q \Rightarrow s, r \Rightarrow s$	$\vdash$	s	<i>Or Elimination</i>
$q \Rightarrow r$	$\vdash$	$\neg q \vee r$	<i>Material Conditional(Or)</i>
$q \Rightarrow r$	$\vdash$	$\neg(q \& \neg r)$	<i>Material Conditional(And)</i>
$q \Rightarrow (r \Rightarrow s)$	$\vdash$	$(q \& r) \Rightarrow s$	<i>Exportation</i>
q	$\vdash$	$q \vee r$	<i>Or Introduction</i>
$q \vee r, q \Rightarrow s, r \Rightarrow s$	$\vdash$	s	<i>Or Elimination</i>

Some important and familiar logical equivalences still hold:

q	$=$	$\neg\neg q$	<i>Double Negation</i>
q	$=$	$q \& q$	<i>Idempotency(And)</i>
q	$=$	$q \vee q$	<i>Idempotency(Or)</i>
$q \& r$	$=$	$r \& q$	<i>Commutativity (And)</i>
$q \vee r$	$=$	$r \vee q$	<i>Commutativity (Or)</i>
$q \& (r \& s)$	$=$	$(q \& r) \& s$	<i>Associativity (And)</i>
$q \vee (r \vee s)$	$=$	$(q \vee r) \vee s$	<i>Associativity (Or)</i>
$\neg(q \& r)$	$=$	$\neg q \vee \neg r$	<i>De Morgan's Law (And)</i>
$\neg(q \vee r)$	$=$	$\neg q \& \neg r$	<i>De Morgan's Law (Or)</i>
$q \Rightarrow r$	$=$	$\neg r \Rightarrow \neg q$	<i>Contraposition</i>

2.4 Conclusion

The improved deductive power of LM_4 relative to paraconsistent systems like LP is readily apparent and is sufficient to enable us to turn on its head the methodological maxim which Priest is forced to adopt. Hence we may stick with LM_4 unless we can be completely assured (in some particular context perhaps) that contradictions cannot arise in our knowledge bases. Furthermore, by confining itself to talking about presence values LM_4 permits the parties for whom it is a genuine concern to differ on the question of truth while still using the system to good effect.

The operator semantics of LM_4 arise naturally as the working out of classical precepts in a more complicated environment. In the bivalent situation these precepts produce what we might call LM_2 and this is effectively classical logic² [10]. In the situation for which LM_4 was designed they yield an effective and productive system which should prove useful for A.I. system builders. While LM_4 cannot cure all of the problems faced, for example by CYC, it does provide a safe mechanism for dealing with contradictions either as an alternative to using buttes or within buttes.

² LM_3 is the subject of a separate paper.

References

1. Quine WVO (1948) On what there is. *The Review of Metaphysics* 2:21-28, Reprint in: *From a logical Point of View*, Cambridge, Harvard University Press (1980), p 18
2. Priest G (1979) The logic of paradox. *Journal of Philosophical Logic*, p 226
3. Priest G (1979) The logic of paradox. *Journal of Philosophical Logic*, p 235
4. Lewis D (1982) A logic for equivocators. *Noûs* 16(3):431-441
5. Lenat DB, Feigenbaum EA (1991) On the Thresholds of Knowledge. *Artificial Intelligence* 47:217-222
6. Copeland BJ (1993) *Artificial intelligence: A philosophical introduction*. Blackwell, Oxford, p 120
7. Lemmon EJ (1992) *Beginning Logic*. Hackett, pp 1-83

Appendix: Indicative Results for LM₄

The test results presented below are drawn from two sets of tests: the first was a group standard inference forms, the second were representations of the sequents and exercises which appear in the first few chapters of E.J. Lemmon's, *Beginning Logic* [7]. The results are presented here without pruning.

- Column 2: Indicates if a logical relation of entailment exists from the premise(s) to the conclusion. Where invalid, I have indicated the first position on the presence table where an invalidity was found.
- Column 3: Indicates if a logical relation of entailment exists from the conclusion to the premise(s). Where invalid, I have indicated the first position on the presence table where an invalidity was found.
- Column 4: '=' indicates the relation of logical equivalence exists between the premise(s) and the conclusion.
- Column 5: The representation of the sequent which was tested.

NB: Although it does not occur in any of the sequents tested below, it should be possible to get a sequent for which the premise(s) entails the conclusion and vice versa but where the premise(s) and conclusion have distinct presence tables. Such a situation could not occur in classical logic (LM₂).

Secondly, while not being a relevance logic, LM₄ does stop a number of the sequents which would concern Relevance Logicians like Lemmon 050 ($q \mid - r \Rightarrow q$).

Table 2.7. Indicative results for LM₄

Ex Contradictione Quodlibet	Invalid when q=e r=a	Invalid when q=a r=e	Dif $(q \ \& \ \neg q) \vdash \neg r$
Disjunctive Syllogism	Invalid when q=e r=e	Invalid when q=d r=a	Dif $((q) \ \& \ \neg q \vee r) \vdash r$
Conditional Proof	Invalid when q=a r=e	Invalid when q=d r=a	Dif $((q) \ \& \ r) \vdash q \Rightarrow r$
And Elimination	Valid	Invalid when q=a r=e	Dif $(q \ \& \ r) \vdash q$
And Introduction	Valid	Valid	= $((q) \ \& \ r) \vdash q \ \& \ r$
Associative Laws (&)	Valid	Valid	= $(q \ \& \ (r \ \& \ s)) \vdash (q \ \& \ r) \ \& \ s$
Associative Laws (V)	Valid	Valid	= $(q \vee (r \vee s)) \vdash (q \vee r) \vee s$
Commutative Laws (&)	Valid	Valid	= $(q \ \& \ r) \vdash r \ \& \ q$
Commutative Laws (V)	Valid	Valid	= $(q \vee r) \vdash r \vee q$
Contraposition	Valid	Valid	= $(q \Rightarrow r) \vdash \neg r \Rightarrow \neg q$
Double Negation	Valid	Valid	= $(q) \vdash \neg \neg q$
Distributive Laws (&)	Valid	Valid	= $(q \ \& \ (r \vee s)) \vdash (q \ \& \ r) \vee (q \ \& \ s)$
Distributive Laws (V)	Valid	Valid	= $(q \vee (r \ \& \ s)) \vdash (q \vee r) \ \& \ (q \vee s)$
Idempotency (&)	Valid	Valid	= $(q) \vdash q \ \& \ q$
Idempotency (V)	Valid	Valid	= $(q) \vdash q \vee q$
Material Conditional (1)	Valid	Invalid when q=a r=e	Dif $(q \Rightarrow r) \vdash \neg q \vee r$
Material Conditional (2)	Valid	Invalid when q=a r=e	Dif $(q \Rightarrow r) \vdash \neg(q \ \& \ \neg r)$
De Morgan's Laws (&)	Valid	Valid	= $(\neg(q \ \& \ r)) \vdash \neg q \vee \neg r$
De Morgan's Laws (V)	Valid	Valid	= $(\neg(q \vee r)) \vdash \neg q \ \& \ \neg r$
Modus Ponens	Valid	Invalid when q=a r=e	Dif $((q) \ \& \ q \Rightarrow r) \vdash r$
Modus Tollens	Valid	Invalid when q=d r=a	Dif $((q \Rightarrow r) \ \& \ \neg r) \vdash \neg q$
Or Elimination	Valid	Invalid when q=a r=a s=e	Dif $((((q \vee r) \ \& \ q \Rightarrow s) \ \& \ r \Rightarrow s) \vdash s$
Or Introduction	Valid	Invalid when q=d	Dif $(q) \vdash q \vee r$

		$r=a$	
Reductio Ad Absurdum	Valid	Invalid when $q=e$	Dif $(q \Rightarrow (r \ \& \ \neg r)) \mid - \neg q$
Transitivity	Valid	$r=a$ Invalid when $q=a$	Dif $((q \Rightarrow r) \ \& \ r \Rightarrow s) \mid - q \Rightarrow s$
Exportation	Valid	$r=e \ s=a$ Invalid when $q=e$	Dif $(q \Rightarrow (r \Rightarrow s)) \mid - (q \ \& \ r) \Rightarrow s$
Lemmon 001	Valid	$r=i \ s=d$ Invalid when $q=a$	Dif $((q \Rightarrow r) \ \& \ q) \mid - r$
Lemmon 002	Valid	$r=e$ Invalid when $q=a$	Dif $((\neg r \Rightarrow (\neg q \Rightarrow r)) \ \& \ \neg r) \mid - \neg q \Rightarrow r$
Lemmon 003	Valid	$r=a$ Invalid when $q=a$	Dif $((((q \Rightarrow r) \ \& \ r \Rightarrow s) \ \& \ q) \mid - s$
Lemmon 004	Valid	$r=a \ s=e$ Invalid when $q=a$	Dif $((((q \Rightarrow (r \Rightarrow s)) \ \& \ q \Rightarrow r) \ \& \ q) \mid - s$
Lemmon 005	Valid	$r=a \ s=e$ Invalid when $q=d$	Dif $((q \Rightarrow r) \ \& \ \neg r) \mid - \neg q$
Lemmon 006	Valid	$r=a$ Invalid when $q=a$	Dif $((((q \Rightarrow (r \Rightarrow s)) \ \& \ q) \ \& \ \neg s) \mid - \neg r$
Lemmon 007	Valid	$r=e \ s=a$ Invalid when $q=d$	Dif $((q \Rightarrow \neg r) \ \& \ r) \mid - \neg q$
Lemmon 008	Valid	$r=e$ Invalid when $q=a$	Dif $((\neg q \Rightarrow r) \ \& \ \neg r) \mid - q$
Lemmon 009	Valid	$r=a$ Valid	$= (q \Rightarrow r) \mid - \neg r \Rightarrow \neg q$
Lemmon 010	Valid	Valid	$= (q \Rightarrow (r \Rightarrow s)) \mid - r \Rightarrow (q \Rightarrow s)$
Lemmon 011	Valid	$r=e \ s=d$ Invalid when $q=a$	Dif $(r \Rightarrow s) \mid - (\neg r \Rightarrow \neg q) \Rightarrow (q \Rightarrow s)$
Lemmon 012	Valid	Valid	$= ((q) \ \& \ r) \mid - q \ \& \ r$
Lemmon 013	Invalid when $q=e$	$r=i \ s=d$ Valid	Dif $((q \ \& \ r) \Rightarrow s) \mid - q \Rightarrow (r \Rightarrow s)$
Lemmon 014	Valid	Invalid when $q=a$	Dif $(q \ \& \ r) \mid - q$
Lemmon 015	Valid	$r=e$ Invalid when $q=d$	Dif $(q \ \& \ r) \mid - r$

		r=a	
Lemmon 016	Valid	Invalid when q=e r=i s=d	Dif $(q \Rightarrow (r \Rightarrow s)) \vdash (q \& r) \Rightarrow s$
Lemmon 017	Valid	Valid	= $(q \& r) \vdash r \& q$
Lemmon 018	Invalid when q=e r=a s=a	Invalid when q=d r=a s=d	Dif $(r \Rightarrow s) \vdash (q \& r) \Rightarrow (q \& s)$
Lemmon 019	Valid	Valid	= $(q \vee r) \vdash r \vee q$
Lemmon 020	Invalid when q=e r=e s=d	Invalid when q=a r=a s=d	Dif $(r \Rightarrow s) \vdash (q \vee r) \Rightarrow (q \vee s)$
Lemmon 021	Valid	Valid	= $(s \vee (q \vee r)) \vdash q \vee (s \vee r)$
Lemmon 022	Valid	Invalid when q=e r=a	Dif $((q \Rightarrow r) \& q \Rightarrow \neg r) \vdash \neg q$
Lemmon 023	Valid	Invalid when q=e r=a	Dif $(q \Rightarrow \neg q) \vdash \neg q$
Lemmon 024	Valid	Valid	= $(q \Rightarrow r) \vdash r \Rightarrow q$
Lemmon 025	Valid	Invalid when q=a r=e	Dif $((q) \& q=r) \vdash r$
Lemmon 026	Valid	Invalid when q=a r=e s=a	Dif $((q=r) \& r=s) \vdash q=s$
Lemmon 027	Valid	Invalid when q=e r=a	Dif $((q \& r)=q) \vdash q \Rightarrow r$
Lemmon 028	Valid	Invalid when q=a r=e	Dif $(q \& (q=r)) \vdash q \& r$
Lemmon 029	Valid	Valid	= $(q) \vdash q$
Lemmon 030a	Invalid when q=e r=i s=d	Valid	Dif $((q \& r) \Rightarrow s) \vdash q \Rightarrow (r \Rightarrow s)$
Lemmon 030b	Valid	Invalid when q=e r=i s=d	Dif $(q \Rightarrow (r \Rightarrow s)) \vdash (q \& r) \Rightarrow s$
Lemmon 031a	Valid	Valid	= $(q \& (q \vee r)) \vdash q$
Lemmon 031b	Valid	Valid	= $(q) \vdash q \& (q \vee r)$
Lemmon 032a	Valid	Valid	= $(q \vee (q \& r)) \vdash q$
Lemmon 032b	Valid	Valid	= $(q) \vdash q \vee (q \& r)$
Lemmon 033a	Valid	Valid	= $(q \vee q) \vdash q$
Lemmon 033b	Valid	Valid	= $(q) \vdash q \vee q$
Lemmon 034a	Invalid when q=e	Invalid when q=d	Dif $((q) \& \neg(q \& r)) \vdash \neg r$

	$r=a$	$r=e$	
Lemmon 034b	Invalid when $q=d$	Invalid when $q=e$	Dif $(\neg r) \vdash q \ \& \ (\neg(q \ \& \ r))$
Lemmon 035a	$r=e$ Valid	$r=a$ Invalid when $q=a$	Dif $(q \Rightarrow r) \vdash \neg(q \ \& \ \neg r)$
Lemmon 035b	$r=e$ Invalid when $q=a$	Valid	Dif $(\neg(q \ \& \ \neg r)) \vdash q \Rightarrow r$
Lemmon 036a	Valid	Valid	$= (q \vee r) \vdash \neg(\neg q \ \& \ \neg r)$
Lemmon 036b	Valid	Valid	$= (\neg(\neg q \ \& \ \neg r)) \vdash q \vee r$
Lemmon 038	Invalid when $q=e$		Dif $\vdash q \Rightarrow q$
Lemmon 039	Invalid when $q=e$		Dif $\vdash q \Rightarrow \neg\neg q$
Lemmon 040	Invalid when $q=e$		Dif $\vdash (\neg\neg q) \Rightarrow q$
Lemmon 041	Invalid when $q=d$		Dif $\vdash q \ \& \ (r \Rightarrow q)$
Lemmon 042	$r=a$ Valid		Dif $\vdash (q \Rightarrow r) \Rightarrow (\neg r \Rightarrow \neg q)$
Lemmon 043	Valid		Dif $\vdash (q \Rightarrow (r \Rightarrow s)) \Rightarrow ((q \Rightarrow r) \Rightarrow (q \Rightarrow s))$
Lemmon 044	Valid		Dif $\vdash q \vee \neg q$
Lemmon 045	Invalid when $q=e$	Valid	Dif $(q) \vdash (q \ \& \ r) \vee (q \ \& \ \neg r)$
Lemmon 046	$r=i$ Invalid when $q=d$	Valid	Dif $(q \Rightarrow r) \vdash q \ \& \ (r=q)$
Lemmon 047a	$r=a$ Valid	Invalid when $q=e$	Dif $((q \ \& \ r)=q) \vdash q \Rightarrow r$
Lemmon 047b	$r=a$ Invalid when $q=e$	$r=a$ Valid	Dif $(q \Rightarrow r) \vdash (q \ \& \ r)=q$
Lemmon 048	$r=a$ Invalid when $q=a$	Valid	Dif $(\neg q \vee r) \vdash q \Rightarrow r$
Lemmon 050	$r=e$ Invalid when $q=e$	Invalid when $q=d$	Dif $(q) \vdash r \Rightarrow q$
Lemmon 051	$r=a$ Invalid when $q=e$	$r=e$ Invalid when $q=a$	Dif $(\neg q) \vdash q \Rightarrow r$
Lemmon 053	$r=e$ Invalid when $q=d$	$r=a$ Invalid when $q=a$	Dif $((\neg r) \ \& \ q \vee r) \vdash q$

	$r=e$	$r=a$	
Lemmon 054	Invalid when $q=e$		Dif $\neg (q \Rightarrow r) \vee (r \Rightarrow q)$
Lemmon 055	Valid $r=e$	Invalid when $q=d$ $r=a$	Dif $((q \Rightarrow r) \& \neg r) \vdash \neg q$
Lemmon Exercise 1(a)	Valid	Invalid when $q=a$ $r=e$	Dif $((q \Rightarrow (q \Rightarrow r)) \& q) \vdash r$
Lemmon Exercise 1(b)	Valid	Invalid when $q=d$ $r=a$ $s=a$	Dif $((r \Rightarrow (q \Rightarrow s)) \& \neg s) \& r \vdash \neg q$
Lemmon Exercise 1(c)	Valid	Invalid when $q=a$ $r=e$	Dif $((q \Rightarrow \neg r) \& q) \vdash r$
Lemmon Exercise 1(d)	Valid	Invalid when $q=a$ $r=e$	Dif $((\neg r \Rightarrow q) \& \neg q) \vdash \neg r$
Lemmon Exercise 1(e)	Valid	Invalid when $q=a$ $r=e$	Dif $((\neg q \Rightarrow \neg r) \& r) \vdash q$
Lemmon Exercise 1(f)	Valid	Valid	$= (q \Rightarrow \neg r) \vdash r \Rightarrow \neg q$
Lemmon Exercise 1(g)	Valid	Valid	$= (\neg q \Rightarrow r) \vdash \neg r \Rightarrow q$
Lemmon Exercise 1(h)	Valid	Valid	$= (\neg q \Rightarrow \neg r) \vdash r \Rightarrow q$
Lemmon Exercise 1(i)	Valid	Invalid when $q=a$ $r=e$ $s=a$	Dif $((q \Rightarrow r) \& r \Rightarrow s) \vdash q \Rightarrow s$
Lemmon Exercise 1(j)	Valid	Invalid when $q=a$ $r=e$ $s=d$	Dif $(q \Rightarrow (r \Rightarrow s)) \vdash (q \Rightarrow r) \Rightarrow (q \Rightarrow s)$
Lemmon Exercise 1(l)	Valid	Invalid when $q=a$ $r=e$ $s=a$	Dif $(q \Rightarrow r) \vdash (r \Rightarrow s) \Rightarrow (q \Rightarrow s)$
Lemmon Exercise 1(m)	Valid	Invalid when $q=d$ $r=a$	Dif $(q) \vdash (q \Rightarrow r) \Rightarrow r$
Lemmon Exercise 1(n)	Valid	Invalid when $q=d$ $r=a$ $s=a$	Dif $(q) \vdash (\neg(r \Rightarrow s) \Rightarrow \neg q) \Rightarrow (\neg s \Rightarrow \neg r)$
Lemmon Exercise 1 p.27-8 (a)	Invalid when $q=a$ $r=e$	Invalid when $q=d$ $r=e$	Dif $(q) \vdash r \Rightarrow (q \& r)$
Lemmon Exercise 1 p.27-8 (b)	Valid	Valid	$= (q \& (r \& s)) \vdash r \& (q \& s)$
Lemmon Exercise 1 p.27-8 (c)	Valid	Valid	$= ((q \Rightarrow r) \& (q \Rightarrow s)) \vdash q \Rightarrow (s \& r)$
Lemmon Exercise 1 p.27-8 (d)	Valid	Invalid when $q=a$	Dif $(r) \vdash q \vee r$

		$r=e$	
Lemmon Exercise 1 p.27-8 (e)	Valid	Invalid when $q=a$	Dif $(q \& r) \vdash q \vee r$
		$r=e$	
Lemmon Exercise 1 p.27-8 (f)	Valid	Valid	$= ((q \Rightarrow s) \& (r \Rightarrow s)) \vdash (q \vee r) \Rightarrow s$
Lemmon Exercise 1 p.27-8 (j)	Valid	Invalid when $q=e$	Dif $(\neg q \Rightarrow q) \vdash q$
Lemmon Exercise 1 p.33 (a)	Valid	Invalid when $q=a$	Dif $((r) \& q=r) \vdash q$
		$r=e$	
Lemmon Exercise 1 p.33 (b)	Valid	Valid	$= ((q \Rightarrow r) \& r \Rightarrow q) \vdash q=r$
Lemmon Exercise 1 p.33 (c)	Valid	Valid	$= (q=r) \vdash \neg q=\neg r$
Lemmon Exercise 1 p.33 (d)	Valid	Valid	$= (\neg q=\neg r) \vdash q=r$
Lemmon Exercise 1 p.33 (e)	Invalid when $q=a$	Invalid when $q=d$	Dif $((q \vee r)=q) \vdash q \Rightarrow r$
	$r=e$	$r=a$	
Lemmon Exercise 1 p.33 (f)	Valid	Invalid when $q=a$	Dif $((q=\neg r) \& r=\neg s) \vdash q=s$
		$r=a \ s=a$	
Lemmon Exercise 1 p.41 (a)	Valid	Valid	$= (q \vee r) \vdash q \vee r$
Lemmon Exercise 1 p.41 (b-1)	Valid	Valid	$= (q \& q) \vdash q$
Lemmon Exercise 1 p.41 (b-2)	Valid	Valid	$= (q) \vdash q \& q$
Lemmon Exercise 1 p.41 (c-1)	Valid	Valid	$= (q \& (r \vee s)) \vdash (q \& r) \vee (q \& s)$
Lemmon Exercise 1 p.41 (c-2)	Valid	Valid	$= ((q \& r) \vee (q \& s)) \vdash q \& (r \vee s)$
Lemmon Exercise 1 p.41 (d-1)	Valid	Valid	$= (q \vee (r \& s)) \vdash (q \vee r) \& (q \vee s)$
Lemmon Exercise 1 p.41 (d-2)	Valid	Valid	$= ((q \vee r) \& (q \vee s)) \vdash q \vee (r \& s)$
Lemmon Exercise 1 p.41 (e-1)	Valid	Invalid when $q=e$	Dif $(q \& r) \vdash \neg(q \Rightarrow \neg r)$
		$r=i$	
Lemmon Exercise 1 p.41 (e-2)	Invalid when $q=e$	Valid	Dif $(\neg(q \Rightarrow \neg r)) \vdash q \& r$
	$r=i$		
Lemmon Exercise 1 p.41 (f-1)	Valid	Invalid when $q=a$	Dif $(\neg(q \vee r)) \vdash \neg q \vee \neg r$
		$r=e$	
Lemmon Exercise 1 p.41 (f-2)	Invalid when $q=a$	Valid	Dif $(\neg q \vee \neg r) \vdash \neg(q \vee r)$
	$r=e$		
Lemmon Exercise 1 p.41 (g-1)	Valid	Valid	$= (\neg q \& \neg r) \vdash \neg(q \vee r)$
Lemmon Exercise 1 p.41 (g-2)	Valid	Valid	$= (\neg(q \vee r)) \vdash \neg q \& \neg r$
Lemmon Exercise 1 p.41 (h-1)	Valid	Valid	$= (q \& r) \vdash \neg(\neg q \vee \neg r)$

Lemmon Exercise 1 p.41 (h-2)	Valid	Valid	=	$(\neg(\neg q \vee \neg r)) \vdash q \& r$
Lemmon Exercise 1 p.41 (i-1)	Valid	Invalid when $q=a$ $r=e$	Dif	$(q \Rightarrow r) \vdash \neg q \vee r$
Lemmon Exercise 1 p.41 (j-1)	Valid	Invalid when $q=d$ $r=e$	Dif	$(\neg q \Rightarrow r) \vdash q \vee r$
Lemmon Exercise 1 p.41 (j-2)	Invalid when $q=d$ $r=e$	Valid	Dif	$(q \vee r) \vdash \neg q \Rightarrow r$
Lemmon Exercise 1 p.62 (a)	Valid		Dif	$\vdash (r \Rightarrow s) \Rightarrow ((q \Rightarrow r) \Rightarrow (q \Rightarrow s))$
Lemmon Exercise 1 p.62 (b)	Invalid when $q=a$ $r=e$		Dif	$\vdash q \Rightarrow (r \Rightarrow (q \& r))$
Lemmon Exercise 1 p.62 (c)	Valid		Dif	$\vdash (q \Rightarrow s) \Rightarrow ((r \Rightarrow s) \Rightarrow ((q \vee r) \Rightarrow s))$
Lemmon Exercise 1 p.62 (d)	Valid		Dif	$\vdash ((q \Rightarrow r) \& \neg r) \Rightarrow \neg q$
Lemmon Exercise 1 p.62 (e)	Valid		Dif	$\vdash (\neg q \Rightarrow q) \Rightarrow q$
Lemmon Exercise 2 p.63 (a)	Valid	Invalid when $q=a$ $r=e$	Dif	$((q \Rightarrow r) \Rightarrow q) \& q \Rightarrow r \vdash q$

Chapter 3

A cooperative distributed query-processing approach

Saeed K. Rahimi, Frank S. Haug

University of St. Thomas, Opus College of Business, 2115 Summit Ave.,
St. Paul, MN 55105, USA

Abstract. Although Distributed Database Management Systems (DDBMS) have been studied for many years there has been little or no emphasis that these studies have put on dynamic systems where database systems can join and depart a group of co-operating DBMS systems. We introduce a new approach to distributed query processing that allows local systems to freely join and leave a distributed system. We achieve this by eliminating the tight synchronization of the global data dictionary and local data dictionaries.

Keywords. Distributed Query Processing, Global Data Dictionary (GDD), Distributed Database Systems (DDBMS), Structured Query Language (SQL), eXtensible Markup Language (XML)

1.1 Introduction

A centralized database management system (DBMS) is a complex software program that allows an enterprise to control its data on a single machine. Recent trend in mergers and acquisitions has resulted in organizations owning more than one DBMS, not all of which have come from the same vendor. It is not too uncommon for an enterprise to own a mainframe database management system from IBM DB2 and a few other departmental databases such as Oracle and Microsoft SQL Server. To control the in-

formation in such an environment one uses a Distributed Database Management System (DDBMS) (See Figure 3.1) [1].

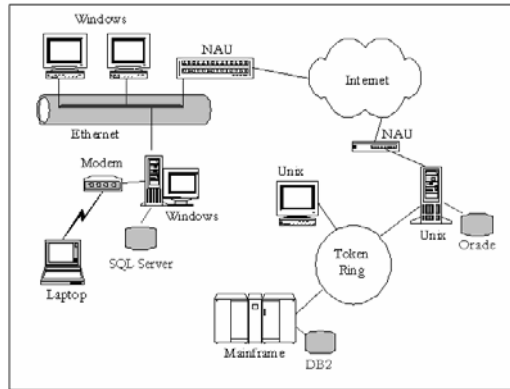


Fig. 3.1. A Traditional DDBMS

A client computer in such a system issues an SQL request to access information that is distributed across the computers in the network. The fact that required data items are distributed and stored on different computers is hidden from the user by the DDBMS system. In other words, the users of distributed data issue their queries as if the system only had one database, was controlled by one database management system (DBMS) and resided on one computer. In order for this to work, a DDBMS system must provide for a set of transparencies.

3.1.1 Transparency types

There are many transparency types that a distributed database system should support. The more transparencies that a DDBMS system supports the easier is the system to use.

1. DBMS transparency – the fact that data are controlled by one or more Local Database Management Systems (LDBMS) from different manufacturer is hidden from the user.
2. Location transparency – the fact that data may be stored anywhere in the system and it is not necessarily local is hidden from the user.
3. Distribution transparency – the fact that database tables might be fragmented is hidden from the user.
4. Replication transparency – the fact that there might be more than one copy of a table or a fragment of a table is hidden from the user.

5. Operating system transparency – the fact that different computers in the system are controlled by different operating systems is hidden from the user.
6. Network/communication transparency – the fact that different computers of the system use different communication protocols to talk to each other is hidden from the user.
7. Hardware/architecture transparency – the fact that different computers may have different architecture (32 bit vs. 64 bit machine, for example) is hidden from the user.

3.2 Traditional DDBMS

The key components of a DDBMS system that implements these transparencies are the Global Data Dictionary (GDD) and Global Execution Manager (GEM). The GDD is a data model representing the definition and deployment details for all the global tables and columns [2]. When the user of a distributed system issues a query to access distributed data, the GEM uses the GDD to validate the definition for all the global tables and columns referenced by the query and then fetches the deployment details from GDD. The GEM will then use these details to create a distributed execution plan for accessing each physical table distributed across the network that implements each global table (or each fragment of the global table) required for the query [3, 4].

Although sounding simple and straightforward, the success, efficiency and validity of a given plan depends heavily upon the accuracy and freshness of the deployment details stored in the GDD. As individual LDBMS become active or inactive in the environment, failure to synchronize the deployment details carefully can cause the GEM to build an inaccurate plan even when an alternate LDBMS is available to complete the plan. Even when recovery techniques are used to build alternate plans in the face of failures, this can be an inefficient and complicated process.

3.3 Co-operative and dynamic DDBMS

Our proposal for processing queries in a distributed system is different from traditional approaches that use a GDD and GEM. These approaches depend heavily on the synchronicity of the GDD and the actual data deployment. This synchronicity is hard to maintain and it is time consuming. We propose an approach to distributed query processing in dynamic envi-

ronments in which databases can freely join and leave the servers in an Intranet. Although our approach uses the concept of a GDD, our GDD only stores the global table and column definitions. Our technique for building the distributed execution plan requires the GDD to be synchronized with the global table and column definitions but not the actual local data deployment details. This approach lends itself nicely to peer-to-peer systems [5], development, testing, and research environments where tables' fragmentation, replication and allocation change constantly. It also allows for multiple, alternative deployments to coexist within the same system seamlessly.

3.3.1 Architecture

Our architecture consists of four programmatic components, two metadata schemas, and the local data stores themselves.

3.3.1.1 Client

Clients are user-written components that can be implemented in J2EE [6], Dot Net, Web Services [7], or similar-minded approaches. The client library / framework can be quite minimal. The clients are small, lightweight, and multithreaded. Placing too much logic or communication overhead in the client library, would make it impractical or impossible for users to create the programs they want.

3.3.1.2 Distributed query service provider

The DQSP is deployed as a web service and registered dynamically via UDDI. The DQSP is also a reasonably lightweight component. The DQSP can be deployed as several instances within the intranet, and can be pooled within an application server (like the Dot Net Framework's managed assemblies or a J2EE application server like JBOSS [8]). The DQSP is merely an encapsulation and decoupling of the communication complexities and potentially complex threading required to process queries and their results correctly and efficiently.

3.3.1.3 Data dictionary service provider

The DDSP contains the Global "Interface" Schema (GIS). The GIS replaces the traditional GDD and contains the global table names, global column names, and global data types / nullability details. The definition of this schema is the only thing that needs to be synchronized system-wide. The

deployment and implementation details are not located here; those details are only located with each LQSP.

This component is also responsible for connecting each DQSP with the appropriate set of LQSP components. This can be a trivial task and can be done by requiring the LQSP components to be registered in such a way that the DDSP can find them. This is done via some form of messaging, perhaps using some standard approach (like WSDL and UDDI). This information can be stored in a database or just cached in memory.

3.3.1.4 Local query service provider (LQSP)

This component is responsible for the local query processing and actually connecting to the LDBMS. Each LQSP would use a schema that connects the GIS with the Database, Location, Fragmentation, and Replication transparency details for all pieces located at this component. This schema is called the Local Deployment Schema (LDS).

3.3.2 Distributed execution plan strategy

The execution plans devised in our approach are no different than those in a traditional system, the only difference is how we actually build and execute them. In general, a distributed execution plan is formed by first translating the global query from SQL into a fully-qualified relational algebra expression. Next, we replace each global table with the set of all distributed tables (or fragments) that represent it and the joins and unions required to reconstruct the global tables from the fragments. The operations can be rearranged to ensure that each operation can be executed within a single LQSP if possible.

At this point, we have a plan suitable for execution, but not an optimal plan. If we look at the relational algebra expression, it is easy to see how things can be rearranged to optimize the query. By “pushing” the projection and selection operations “down” (closer to the actual tables) and rearranging the order in which things are joined we can improve the performance by reducing the number and size of the rows produced by each intermediate step and reduce the number of times that results need to be combined from multiple LQSP sites. Lastly, when the results from more than one LQSP do need to be combined (using union, join, or semi-join operations) we can use the general guiding principle of sending the smaller result set to the LQSP with the larger result set. This minimizes the communication cost. There can be more complex strategies used for special cases but this overall approach is sufficient for most simple queries.

3.3.3 Communication Overview

Figure 3.2 depicts the overall communication of our system components.

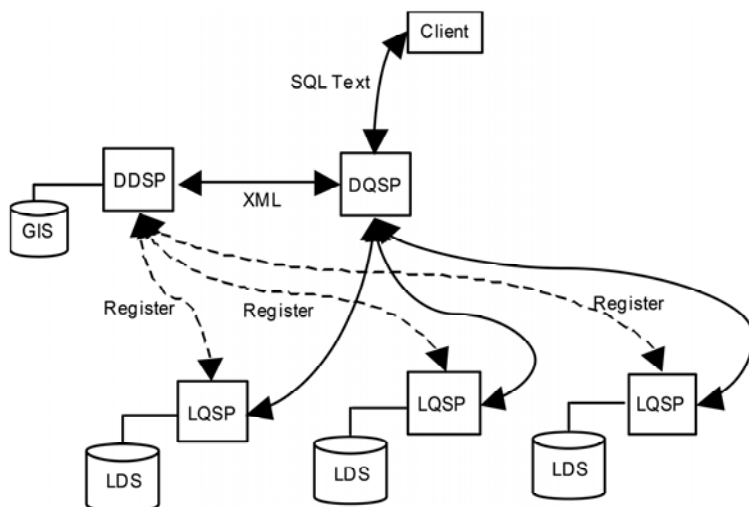


Fig. 3.2. Communication in a P2P DDBMS

1. A Client initializes the library, which looks up the DQSP to use.
2. A Client sends a query to the DQSP as raw SQL text.
3. The DQSP parses the SQL and builds a simple XML [9] version of the query.
4. The DQSP sends the XML query to the DDSP.
5. The DDSP disambiguates and validates the query. It then builds a list of candidate LQSP's that could potentially fulfill this query.
6. The DDSP returns the candidate list and fully-qualified query to the DQSP. The DQSP sends this query request to each candidate LQSP.
7. Each candidate LQSP will look at the request and look at their LDS determining what (if any) part it could answer along with a "self-ranking" indicating some measure of the "coverage" or "efficiency" with which it can do things. For example, if a LQSP had an instance of all tables versus only some tables versus only some fragment(s) of some table(s).
8. The system needs to elect the actual LQSP from the candidate LQSP in the list. This is controlled and coordinated by the DQSP by evaluating the "self-ranking" information from each responding LQSP.
9. DQSP uses a timeout-based approach and waits for the "best" answer available and then closes the election once a "good enough" group of candidates has responded.

10. Now the DQSP can tell the elected LQSP to perform the processing. The DQSP sends the list of elected LQSP along with the self-ranking information returned by each to each elected LQSP. Each LQSP can then use the same information and algorithm to build their portion of the execution plan. This means that each LQSP determines its local plan, determines which component it should send its results to, and determines which components it should expect to receive results from. When results are received, they are merged and passed to the appropriate LQSP or cached for retrieval by the client. The LQSP containing the final results will message the DQSP when they are ready.
11. The DQSP receives notification of the query completion status (success or failure) and messages the client with the result.
12. The client receives notification and can then send additional messages to retrieve the result in multiple chunks or in its entirety.

3.3.3.1 DQSP process

1. A DQSP initializes, which looks up the DDSP to use.
2. The DQSP receives a SQL query from a client.
3. The DQSP parses the SQL into simple XML.
4. The DQSP sends the XML query to the DDSP.
5. The DQSP receives acknowledgement that the SQL is valid along with a LQSP candidate list and fully-qualified query, or an error if there was a problem with the query (in which case it then sends the error back to the client.)
6. The DQSP sends query request messages to the LQSP candidate list.
7. The DQSP receives responses from the candidates and determines if it has enough “good enough” candidates responding within the timeout. If it does not then it sends the error back to the client.
8. The DQSP sends an “OK” to all the elected candidates and processes the results (merging if necessary).
9. The DQSP returns the results to the client.

3.3.3.2 DDSP process

1. A DDSP initializes, which connects to the GIS and registers the DDSP for use.
2. The DQSP registers and un-registers LQSP components as needed.
3. The DDSP receives the simple XML query. It validates this against the GIS, and makes it fully-qualified. If it is not valid, or if it is ambiguous, it returns an error to the client.

4. The DDSP looks at the registered LQSP and the fully-qualified query. It builds a list of candidate LQSP and returns the fully-qualified XML query along with the list of candidate LQSP to the DQSP.

3.3.3.3 LQSP process

1. A LQSP initializes, which looks up the DDSP to use.
2. The LQSP registers itself with the DDSP.
3. The LQSP receives global query (in XML) from DQSP. It uses the statistics and deployment information in the LDS to determine what type of coverage it can provide for this query. As part of this process, it builds up a local execution plan.
4. The LQSP sends its coverage and related statistics (such as number of rows, number of columns, types of fragmentation it has, etc.) to the DQSP.
5. The LQSP receives the notification that it has been elected to process a query. It processes the local plan for the query and sends its results to DQSP or other LQSP's as per plan.

3.3.4 Query processing example

Consider a simple global schema containing two tables. The Emp table is vertically fragmented to separate sensitive salary details from the less sensitive employee name and job title columns while the Dept table is horizontally fragmented by location (See Figure 3.3). For example, suppose Dept_HF_1 were defined to contain all rows with Loc in ('MN', 'IA', 'WI') while Dept_HF2 will contain all rows with any other value for Loc.

```

Global Table: Emp(Eno, Name, Title, Salary, Dno)
Fragment: Emp_VF1(Eno, Salary, Dno)
Fragment: Emp_VF2(Eno, Name, Title)
Global Table: Dept (Dname, Dno, Budget, Loc)
Fragment: Dept_HF1 (Dname, Dno, Budget, Loc)
Fragment: Dept_HF2 (Dname, Dno, Budget, Loc)

```

Fig. 3.3. Example fragmentation

For this example, there are three LQSP sites (See Figure 3.4). The Emp_VF1 fragment containing the salary information is deployed within the LQSP at Site-A. The LQSP at Site-B contains the other vertical fragment (Emp_VF2) and one of the horizontal fragments (Dept_HF1) while Site-C contains the other horizontal fragment (Dept_HF2).

Site A stores Emp_VF1
Site B stores Emp_VF2 and Dept_HF1
Site C stores Dept_HF2

Fig. 3.4. Example deployment

Suppose a client submitted the SQL query shown in Figure 3.5’s column 1. The client would send this SQL string to the DQSP. The DQSP would then translate this query into relational algebra, encode it in XML and pass it to the DDSP for verification and disambiguation. The DDSP would verify that all the global tables and columns referenced in the query were valid and generates the query in column.

Original Global Query	Fully Qualified Global Query
Select Name, Salary, Budget From Emp, Dept Where Emp.Dno = Dept.Dno AND Title = ‘Engineer’;	Select T1.Name, T1.Salary, T2.Budget From Emp T1, Dept T2 Where T1.Dno = T2.Dno AND T1.Title = ‘Engineer’;

Fig. 3.5. Example global query

Using the current list of registered LQSP components, DDSP would build a list of candidate sites (here Site-A, Site-B, and Site-C). The fully-qualified query and candidate site list would be returned to the DQSP. The DQSP would send the fully-qualified query to all candidate sites and await their responses.

Upon receiving the query each LQSP would then form a local plan and determine how well it can satisfy the query. This self-ranking would be based upon the fragmentation plan and actual fragments deployed at each LQSP site. Each LQSP would also use locally gathered and cached statistics to estimate the size (number of rows and row width) for each result produced at that LQSP.

In this example, the local execution plan built by the LQSP at Site-A would return the entire fragment. The coverage information would indicate that the Emp_VF1 fragment was deployed here by including a unique plan id (a GUID) and bit mask flag for the Emp table. The size estimate would use the size of the columns and locally gathered statistics to indicate the expected size for the result. In this case it is merely the total number of rows and sum of the column widths.

The local execution plan built by the LQSP at Site-B would use two separate commands and produce two separate results. The first command would select only rows with Title = ‘Engineer’ from the Emp_VF2 fragment and project the Eno and Name columns as the final result. The second command would project the Dno and Budget columns from the

Dept_HF1 fragment as the final result. The coverage information returned would indicate that the EMP_VF2 fragment was deployed here by including the same plan GUID as Site-A, and a complimentary mask value. Similarly, the second command would indicate a different plan GUID for the Dept fragmentation and a different bit mask value for the Dept_HF1 horizontal fragment deployed here. The size estimate for each result would be included in the self-ranking response from Site-B. In this case, the first result would use locally cached selectivity statistics to estimate the number of rows matching 'Engineer' and the size of the Eno and Name columns. The results from the second command would use the number of rows statistic for the Dept_HF1 fragment and the column size for the Dno and Budget columns.

At Site-C, the LQSP would build a plan containing a single command, similar to the second command at Site-B. In other words, it would project the Dno and Budget columns from the Dept_HF2 fragment. The coverage information from this command would use the same plan GUID as the second command at Site-B, and a complimentary bit mask value. The size estimate would use the number of rows in the Dept_HF2 fragment and the size of the Dno and Budget columns.

When different fragments from the same fragmentation plan for the same table are deployed at different LQPS sites, the LDS must be populated with the deployment information required. In other words, the LDS at each site contains the fragmentation plan criteria, a plan GUID unique to this plan, a bit mask value representing the fragment deployed here, and a bit mask value representing the reconstructed table. In our example, the Emp coverage information at Site-A could have a bit mask value of 0x01, at Site-B a value of 0x02 and both sites would know that the reconstructed table requires a bit mask of 0x03. The GUID stored for the Emp fragmentation plan at each site would be the same value. Similarly, the Dept fragmentation plan deployed at Site-B and Site-C would use a different GUID than the Emp fragmentation plan and bit mask values for each fragment and the reconstructed Dept table based upon its plan.

All of the coverage and size estimate information from each LQSP is returned to the DQSP. The DQSP uses the actual response along with the coverage and size details it contains to elect the official list of sites to be used. In this case, Site-A, Site-B, and Site-C are all available and useful. Therefore, the DQSP elects them and sends another message to each elected LQSP. This message would include the list of all elected sites as well as the coverage and size statistics that each elected site had returned prior to the election.

Using this information, the local plan at Site-A would know that it must join with results from the first command at Site-B in order to reconstruct

the Emp table. Similarly, Site-B and Site-C know that the results of the second command at Site-B and the results of the command at Site-C must be unioned together to reconstruct the Dept table. All three sites know that the reconstructed Emp table and Dept table need to be joined and that the Name, Salary, and Budget columns need to be projected from this result in order to satisfy the original query.

Since the elected LQSP at each site has the same algorithm, same fully-qualified query, and same coverage and size statistics, they can each come to the same decision with respect to where results (both the intermediate and final) are sent. For our example, suppose the first result from Site-B is smaller than the result for Site-A's local plan. Site-B would then know to send its result to Site-A, and Site-A would know to expect it. Similarly, suppose that the result from Site-C is smaller than the second result at Site-B. Both sites would expect the results from Site-C to be sent to Site-B for the union operation. Lastly, using the size of the results for the two Emp fragments about to be joined, all sites can estimate the size of the result, it will be the same number of rows as the smaller of the two fragments (in other words the same number of rows as the first result at Site-B) and the size of the Dno, Name, and Salary columns. The size of the union would be the sum of the size for the second result at Site-B and the result at Site-C. Once again, all sites can determine this for themselves and come to the same decision as to where these two results will be joined to produce the final result. In this case, suppose the Emp result is smaller, Site-A would then send its result to Site-B. The result from Site-A would be joined with the result at Site-B and the final result would be ready for the client to process. The LQSP at Site-B would then notify the DQSP that the results are available.

As the client requests rows to be returned from the DQSP, the DQSP can connect to this LQSP and retrieve the rows. Although it is possible to return all the rows in a single message, it is most likely more efficient to have the number of rows returned be a configurable or parameterized value less than this. The client could use the DQSP as a proxy or connect directly to the LQSP containing the results depending upon the physical network topography, security concerns and installation decisions chosen for a given organization.

3.4 Conclusion

Using the GIS and LDS to decouple the deployment details from the global table and column definitions; we can eliminate the need to keep a single

monolithic and synchronized GDD. This separation also reduces the amount of information that needs to be sent to the DDSP when a given LQSP needs to register with the DDBMS. By eliminating the synchronization of deployment details and reducing the size and complexity of information necessary to register a LQSP with the system, it becomes practical for dynamic LQSP registration. Returning the self-ranking coverage and size statistics from each candidate LQSP allows us to build the execution plan in a dynamic and distributed fashion. A byproduct of the separation of deployment details is the ability to support multiple deployment plans for the same global table. Although each LQSP providing fragments for the same table in the same query must use the same plan, the system can choose to use any plan for a given table provided that the LQSP required for that plan all respond to the election process in a timely fashion.

Unlike a traditional DDBMS, there is no bottleneck at the GEM/DEM for query processing because the majority of the work is done in parallel by the LQSP. In a traditional DDBMS, the synchronization of the GDD can cause deadlock and live lock concerns within the GDD (query processing requiring read access while LDBMS activation and deactivation requires write access). In our approach, this is no longer required. The only time we need to update the GIS is when there is either a new global construct or a change to the interface of an existing global construct (for example, adding a new global column to an existing global table, etc.) This is unavoidable, but it is also an event that happens far less frequently than the activation / deactivation of an LQSP or a new deployment plan.

Although other variations could be implemented such as allowing the LQSP to communicate while deciding where to send results, this would increase the communication complexity and cost with only minor potential for improved performance for most typical queries.

References

1. Özsu MT, Valduriez P (1999) Principles of distributed database systems. Second edition, Prentice Hall
2. Larson J, Rahimi S (1984) Tutorial: Distributed database Management. Computer Society Press
3. Kossmann Donald (2000) The state of the art in distributed query processing. ACM computing surveys 32(4):422-469
4. Ceri S, Pelagatti G (1984) Distributed databases - Principles and Systems. McGraw Hill Inc.

5. Ripeanu M (2001) Peer-to-peer architecture case study: Gnutella network. In: Proceedings of International Conference on Peer-to-peer Computing, Linköping Sweden, pp 99-100
6. Singh Inderjeet, Stearns Beth, Johnson Mark, and the Enterprise Team (2002) Designing Web Services with the J2EE(TM) 1.4 Platform: JAX-RPC, SOAP, and XML Technologies (The Java Series). Addison-Wesley
7. Singh Inderjeet, Brydon Sean, Murray Greg, Ramachandran Vijay, Violleau Thierry, Stearns Beth (2004) Designing Web Services with the J2EE(TM) 1.4 Platform: JAX-RPC, SOAP, and XML Technologies. Addison-Wesley
8. Fleury M, Reverbel F (2003) The JBoss Extensible Server. In: Proceedings of the 2003 ACM / IFIP / USENIX International Middleware Conference, pp 344-373
9. Extensible Markup Language (XML), May 2007, <http://www.w3.org/XML/>

Chapter 4

Estimating join and projection selectivity factors

Carlo dell'Aquila, Ezio Lefons, Filippo Tangorra

Dipartimento di Informatica, Università di Bari, Via E. Orabona 4, I-70125
Bari, ITALY, {dellaquila, lefons, tangorra}@di.uniba.it

Abstract. The selectivity factor of relational operations is a critical parameter for determining the cost function of query processing. Good estimates of these parameters allow the optimizers to choose the least expansive path in the query execution. A method for estimating the join and projection selectivity factors based on orthogonal polynomial series is presented. Experimental results on real data are also reported which show the good performance of the approach.

Keywords. Data warehouse, query cost estimation, query optimization, join selectivity, projection selectivity.

4.1 Introduction

With the growing number of large data warehouses for decision support applications, to efficiently execute data queries is becoming increasingly important. The use of data profile yields advantages not only in establishing efficient strategies to minimize access to large data sets in the traditional data base field, but also can be used in activities such as data warehouses in which data are continually collected from different source databases, in data mining applications that require to access large data sets for complex exploratory activity, and in networking management in which decisions are needed on how to move data to minimize the data traffic. Data profiles maintain statistics about data as the shape form of data distribution, data cardinality, the number of distinct values, and so on, in order to

determine accurate estimates of the query counts and selectivity factors. Using this information, the query optimizer parses and analyzes the query, constructs possible access paths, and estimates the cost of each path in order to determine the least expensive one.

In addition, data profiles are also important in data mining applications and data analysis activity, where the user explores various hypotheses on data and may prefer an approximated result to the query in real time, rather than waiting for the exact one. This research field has conducted to the design of systems which involve the estimates of summary data and the execution of aggregate queries with approximate answers derived by the data profile.

To obtain reasonable cost estimates of query execution, the query optimizers need estimates of sizes of the final and intermediate relations involved, based on the selectivity factors (Caudhuri and Narasayya 2001). The selectivity factor corresponds to the fraction of tuples which satisfy the query.

The methods to estimate the actual query selectivity factors are an important research topic in traditional databases. Various methods, such as parametric (Chen and Roussopoulos 1994; Ling and Sun 1999), sampling (Ling and Sun 1995; Chaudhuri *et al.* 2001), histogram (Ioannidis and Poosala 1995), wavelet (Vitter and Wang 1999), and discrete cosine transform (Jiang *et al.* 2006), etc., have been proposed.

We use the analytic approach based on the approximation of the actual multivariate data distribution of attributes by series of orthogonal polynomials (Lefons *et al.* 1995). The method is a special case of the least squares approximation by orthonormal functions and summarizes all the information on the data distribution by few data—the computed coefficients of the polynomial series, or the so-called *Canonical Coefficients* (CC) of the data. The coefficients CC allow the selectivity factors and main data statistics to be easily derived and efficiently computed with no further access to the data warehouse.

According to the good performance in estimating the selectivity factor of the restriction operation in multidimensional environment (Lefons *et al.* 1995), we have applied the analytical approach to estimate the selectivity factor of join and projection operations. Relative to these relational operations, the knowledge of (counts of) distincts of database attributes plays a crucial role in improving the performance of the method in estimating the selectivity factors. The analytic method, opportunely adapted, permits to enrich the descriptive contents of data profile with few metadata relative to the distribution of distances of unique values, in the attribute value ranges.

Here, we describe the application of the analytic method to determine the selectivity factor of join and projection operations using a new method to estimate the attribute distincts.

4.2 Related work

In this Section, we briefly review methods used in estimating the selectivity factor respectively for join and projection relational operations.

Here, “join” refers to the *equijoin* relational operation. When considering the join T between relations R and S , the canonical formula giving the cardinality of join is $card(T) = jp \times card(R) \times card(S)$ where jp is the join selectivity factor and represents the fraction of tuples of the Cartesian product of R and S which satisfy the join condition.

There are several methods for estimating join cardinality (Jiang *et al.* 2006; Bell *et al.* 1989; Ioannidis and Christodoulakis 1993; Mullin 1993; Sun *et al.* 1993; Haas *et al.* 1996). Some methods are based on integrity constraints of the relational schema and provide limits for the join cardinality estimation (e.g., $card(T) \leq card(R \times S)$). These methods do not provide accurate estimations. Other methods assume arbitrarily that join attributes are uniformly distributed and that the number of distinct values of attribute $T.X$ is $\min(dist(R.X), dist(S.Y))$ (Merrett and Otoo 1979). Indeed, it is very rare that attributes are uniformly distributed in real cases and the uniformity assumption leads to pessimistic results when data attributes accord to other distributions such as Normal, Gamma, Zipf, etc.

Others methods have been proposed. In the worst case assumption method, join cardinality is estimated as $card(R) \times card(S)$. The worst case divided 2 method minimizes errors produced by the worst case method. Methods of perfect knowledge does not produce errors but all the occurrences of attributes values must be stored, so it is very expensive in storage and time consuming for updating (Bell *et al.* 1989). Piece-wise uniform method has been proposed and used in (Mullin 1993) to estimate the frequency distribution of join attributes by equal-width histograms. In this approach, the attribute domain is divided into intervals and the number of tuples holding values which fall into each interval is stored. Attribute distribution is approximated by a piece-wise linear curve which interpolates some points by a relative frequency histogram, which is then used to estimate the size of relational query and parameters of intermediate results. Join range is divided into a number of intervals whose scale is function of the number of distinct values within the join range. To determine the number of distinct values, the (number of) distinct values of joining attributes

should be known, but no proposed method estimates them. So, a way to obtain this is to sort join attribute values and to eliminate duplicates. However, this method is impractical due to the high cost of processing and maintenance after updates. The approach presented here estimates the resulting size of the relational join by evaluating the distinct values and using the estimation of actual distribution of join attributes.

The selectivity of projection is the ratio of reduction of cardinalities. In fact, notice that the projection removes not only attributes but also the resulting duplicate tuples.

In general, the approach for estimating the size of a projection consists of the ability of data statistical profile to fit the actual multidimensional data distribution in order to derive estimates of the marginal distribution of the projected attributes.

In (Merrett and Otoo 1979) is described an approach based on the multidimensional histogram divided into equal-sized buckets, assuming uniformity within the buckets. Parametric methods described in (Gardy and Puech 1984) assume that attributes are uniformly distributed and independent each other. Under these assumptions, authors studied the probability distributions of the sizes of the projections testing various hypotheses. The assumption of independence and/or uniformity of attributes result inadequate to correctly represent many actual database instances. However, they simplify and speed the computations of parameters for run-time query optimizer in evaluating access plans. The model presented in (Ciaccia and Maio 1995) assumes that values of attributes determine a time dependent active domain. Others approaches are based on fast ways to estimate distinct values (Gibbons 2001; Haas *et al.* 1995).

The approach described here estimates the resulting size of the relational projection by evaluating distinct values of active domains of the attributes involved in the operation.

4.3 Data distribution approximation by orthogonal series

Let R be a relation of cardinality N and let X be an attribute of R . We assume $dom(X) = [a, b]$ and let $x_1, x_2, \dots, x_N$ be the occurrences of X in R .

The probability density function $g(x)$ of the attribute X is approximated as

$$g(x) = \frac{1}{b-a} \sum_{i=0}^n (2i+1) c_i P_i(x) \quad (4.1)$$

for $x \in X$ and the opportune n . For each $i = 0, 1, \dots, n$, $P_i(x)$ is the Legendre orthogonal polynomial of degree i , and coefficient c_i is the mean value of $P_i(x)$ on the instances of X . That is,

$$c_i = \frac{1}{N} \sum_{x \in X} P_i(x) \quad (4.2)$$

The $c_0, c_1, \dots, c_n$'s are computed with simple recursive formulæ (Lefons *et al.* 1995) and they are the so-called *Canonical Coefficients* of X .

The approximation of the cumulative distribution function $G(x)$ of $g(x)$ is defined as

$$G(x) = \int_a^x g(y) dy = \frac{1}{b-a} \sum_{i=0}^n c_i (P_{i+1}(x) - P_{i-1}(x)) \quad (4.3)$$

Let $I = [x_1, x_2] \subseteq [a, b]$ be a generic query-range of X . We denote with $count(x; I)$ or $N \times percent(x; I)$ the number of tuples of $R.X$ whose x -values belong to the interval I . Then, $count(x; I)$ can be approximated by $N \times (G(x_2) - G(x_1))$.

4.4 Join selectivity estimation

In this Section, we present how to estimate the cardinality of join operation between relations R and S on the respective attributes X and Y using the analytic model described in the Section 4.3.

Let $T = R \bowtie_{X=Y} S$. In order to estimate $card(T) = jp \times card(R) \times card(S)$ or equivalently jp , let $x_1, x_2, \dots, x_{dist(X)}$ and $y_1, y_2, \dots, y_{dist(Y)}$ be the ordered distinct values of attributes X and Y , respectively. Then,

$$card(T) = \sum_{i=1}^{dist(X)} \sum_{j=1}^{dist(Y)} [C_X(x_i) \times C_Y(y_j)]_{x_i = y_j} \quad (4.4)$$

where $C_X(x_i)$ ($C_Y(y_j)$ respectively) denotes the number of the tuples of R (resp., S) such that $X = x_i$ (resp., $Y = y_j$). In general, the estimation of $C_X(x_i)$ and $C_Y(y_j)$ is obtained without scans of R and S , but only by using the aggregate function *percent* (cf., Section 4.3), as

$$\begin{aligned} C_X(x_i) &\cong count(x; I_{x_i}) = card(R) \times percent(x; I_{x_i}) \\ C_Y(y_j) &\cong count(y; I_{y_j}) = card(S) \times percent(y; I_{y_j}) \end{aligned}$$

where $I_{x_i} = [(x_i + x_{i-1})/2, (x_i + x_{i+1})/2]$ and $I_{y_j} = [(y_j + y_{j-1})/2, (y_j + y_{j+1})/2]$. So, the selectivity factor is approximated as follows.

$$j\rho \cong \sum_{i=1}^{dist(X)} \sum_{j=1}^{dist(Y)} \left[percent(x; I_{x_i}) \times percent(y; I_{y_j}) \right]_{x_i=y_j} \quad (4.5)$$

The *percent* value depends on the distinct values of attributes X and Y, which are often unknown. Many researchers suppose that differences (distances) between two adjacent domain values are approximately equal (e.g., Sun *et al.* 1993).

The approach described here does not make hypotheses about distinct values spacing distribution or their joining equivalence. Distinct values are evaluated using the canonical coefficients which are easily computed and updated whenever a new distinct value is inserted into the database attribute.

Let $\{x_1, x_2, \dots, x_d\}$ be the distinct values of attribute X. The canonical coefficients $(d_i)_{1 \leq i \leq n}$ up to degree n , that contain information about how distinct values are spaced, are computed as follows.

$$d_i = \frac{1}{d \times dns} \sum_{j=1}^d \sum_{k=1}^{dns} P_i(x_j - \frac{\delta}{2} + \delta \times rand(0,1)); \quad i = 0,1,\dots,n \quad (4.6)$$

where we suppose that, for each x_j , there are distributed a sufficiently high number dns of random values in the interval $[x_j - \delta/2, x_j + \delta/2]$, with $\delta = (b - a)/card(X)$. We assume that x is an approximation of the distinct value x_i , if it is verified the condition $count(x; I) \approx dns$ in the interval I or, equivalently, $|count(x; I) - dns| \leq \varepsilon$ for $\varepsilon = k \times dns$ ($0 < k < 1$). It follows the algorithm that computes the approximation of distinct values and their number based on the knowledge of canonical coefficient d_i .

Algorithm 4.1: determination of distinct values.

```

d:= 1;  x(d):= a;  x1:= a + δ/2;  x2:= x1 + δ;  eps:= 0.05 × dns;
while (x2 < b) do begin
    I:= [x1, x2];  occ:= count(x; I);
    if eps ≥ |dns - occ| then begin
        d:= d + 1;
        x(d):= (x1 + x2)/2 end
    x1:= x2;  x2:= x1 + δ
end
d:= d + 1;  x(d):= b

```


When Algorithm 4.1 terminates, the value d approximates the number of distinct values of X in $[a, b]$, and the vector $x[1..d]$ contains the approximation of distinct values.

As in computing canonical coefficients up to degree n in $I_j = [x_j - \delta/2, x_j + \delta/2]$ we have distributed dns occurrences, then it is expected that $count(x; I_j) \cong dns$. So, Algorithm 4.1 analyses interval $[a, b]$ searching for sub-interval I which satisfies the condition of the required estimation accuracy at the given confidence level, defined empirically as the fraction 0.05 of the dns quantity in the interval I of amplitude δ .

Series of canonical coefficients $(d_i)_{0 \leq i \leq n}$ for distincts is different from series computed to approximate the distribution of attribute X . In particular, aggregate function *count* in Algorithm 4.1 is calculated using $(d_i)_{0 \leq i \leq n}$.

Assuming that sets $X = \{x_i\}_{0 \leq i \leq dist(X)}$ and $Y = \{y_j\}_{0 \leq j \leq dist(Y)}$ estimated by Algorithm 4.1 are denoted with $\bar{X} = \{\bar{x}_i\}_{0 \leq i \leq dist(X)}$ and $\bar{Y} = \{\bar{y}_j\}_{0 \leq j \leq dist(Y)}$, to compute join selectivity we use, instead of applying (4.5), the following

$$jp \cong \left[\sum_{i=1}^{dist(X)} \sum_{j=1}^{dist(Y)} percent(x; \bar{I}_{\bar{x}_i}) \times percent(y; \bar{I}_{\bar{y}_j}) \right]_{\bar{x}_i \cong \bar{y}_j} \quad (4.7)$$

We assume the equivalence $\bar{x}_i \cong \bar{y}_j$ if $|\bar{x}_i - \bar{y}_j|$ is lower than the mean distance of adjacent approximate distincts respectively of X and Y .

We have performed experiments on a real case considering the relation $R(A, B, C, D, E, F)$ having $m = 104,828$ tuples.

The values of attributes A and B of relation R are integer (with equally-spaced distinct values). The values of attributes C, D, E and F are real numbers.

Table 4.1 reports the join selectivity estimates by comparing our approach, or the CC method, and the histogram approach, or the EW method. In the CC approach, no hypotheses have been made on the number of distinct values and about their spacing in the join range $[a, b]$. In this case, distinct values for the joining attributes X and Y have been estimated using the method discussed in Section 4.2. According to empirical tests, the distinct values of the attributes are approximated using $dns = 50$, $\varepsilon = 0.05 \times dns$ and approximation degree $n = 27$, while the join selectivity factor has been estimated using formula (4.7).

In the EW approach, the number of distinct values for each attribute has been estimated considering the join range divided into equal-width intervals and values in each interval have been considered to be join-equivalent (Ioannidis and Poosala 1995).

The entries in Table 4.1 indicate the join attributes, the actual number of distinct values d on join range $[a, b]$, the number of distinct values d_{cc} estimated using Algorithm 4.1, and the number of distinct values d_{ew} estimated by the EW method. The mean percentage error is measured using the metric

$$M = |jp - \overline{jp}| / jp$$

where jp and $\overline{jp}$ are, respectively, the join selectivity factor and its estimation relative to methods CC and EW.

Table 4.1. Mean percentage errors of join selectivity estimates

X $\bowtie$ Y	d	d_{cc}	d_{ew}	n_{cc}	Mean Percentage Relative Errors	
					CC	EW
A $\bowtie$ A	16	15	18	8	1.04	3.12
				13	0.81	2.80
				27	0.93	2.95
B $\bowtie$ B	16	15	18	8	1.98	0.42
				13	1.70	0.04
				27	1.58	0.08
C $\bowtie$ C	80	179	85	8	22.09	695.94
				13	10.15	821.94
				27	3.09	888.55
D $\bowtie$ D	286	285	298	8	0.90	13.85
				13	8.23	4.62
				27	5.26	1.36
E $\bowtie$ E	253	225	285	8	13.87	34.85
				13	8.97	14.82
				27	5.01	2.11
F $\bowtie$ F	175	174	184	8	12.34	5586.70
				13	5.84	5586.70
				27	4.49	5586.70

For each method, when estimating the cumulative distribution function by canonical coefficients, we have used approximation degrees 5 to 33. However, for the sake of readability, here we only report the mean percentage error measured for $n_{cc} = 8, 13, 27$ using metric M .

The errors obtained using the CC estimates of join selectivity are significantly reduced with respect to the EW ones. In particular, the results of EW are pessimistic for the attributes whose distinct values are not equally-spaced. For these attributes, the mean percentage error is generally lower for CC method than the EW one.

4.5 Projection selectivity estimation

The model of Gardy-Puech (Gardy and Puech 1984), based on the attribute independence and uniform distribution assumptions, is frequently used to estimate the size of projections. Using probabilistic arguments, they derive formulas to compute the l tuples of projection $\pi_Y(R)$ from the m tuples of a relation $R(X)$, with $Y \subseteq X$, as the expected value of all possible randomly generated projections. This value is computed as

$$l = \frac{d_x}{d_y} \left[1 - \left(1 - \frac{d_y}{d_x} \right)^m \right] \quad (4.8)$$

where d_x and d_y are the product of distinct value numbers respectively of attributes X and Y .

We have used the Gardy-Puech formula, but we estimate the distinct values with canonical coefficients using the effective distinct values in the domain (*i.e.*, the *active* domain). In fact, canonical coefficients allow to estimate current distincts for they are updated when a new/old occurrence is added/removed in/from database. For example, if values $(d_i)_{0 \leq i \leq n}$, dns , and δ are stored in the statistical profile at a certain time and the occurrence x' arrives for attribute X in database, then Algorithm 4.2 updates canonical coefficients d_i if it establishes that x' is a new distinct value.

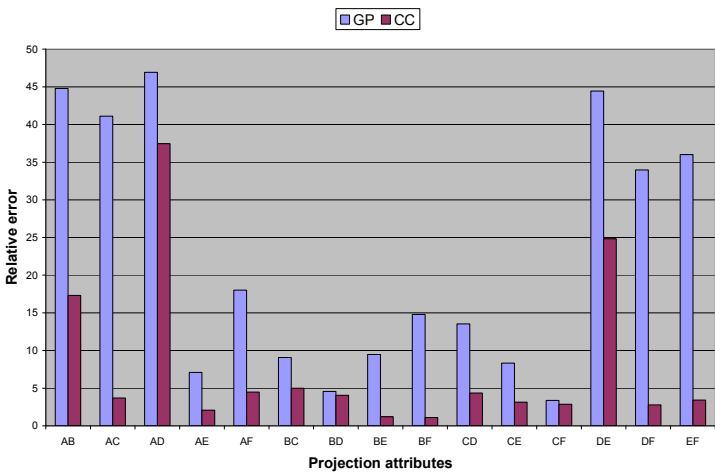
Algorithm 4.2: update of canonical coefficient d_i for distinct values, $0 \leq i \leq n$.

$$\begin{aligned} & I = [x' - \delta / 2, x' + \delta / 2], \\ & \text{if } |d \times dns \times \text{percent}(x; I) - dns| \leq eps \\ & \quad \text{then } x' \text{ already exists in database} \\ & \quad \text{else } x' \text{ is a new distinct value hence } \mathbf{begin} \\ & \quad \quad d_i \times (d \times dns) + \sum_{k=1}^{dns} P_i(x' - \frac{\delta}{2} + \delta \times \text{rand}(0,1)) \\ & \quad d_i := \frac{\quad}{(d+1) \times dns}; \quad i = 0, \dots, n; \\ & \quad d := d + 1 \text{ end} \end{aligned} \quad (4.9)$$

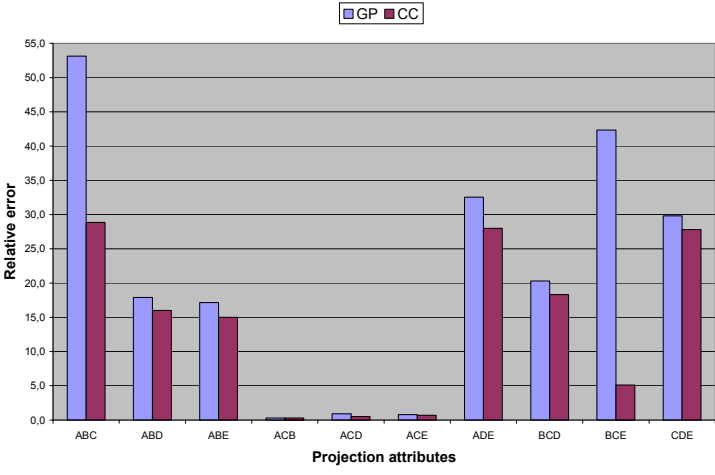
Using the same data and assumptions of the previous Section, Figures 4.1a and 4.1b report the performance of the projection selectivity respectively in the bidimensional and tridimensional cases, by comparing the method of Gardy-Puech (GP) to the analytic method (CC).

The errors obtained in applying the analytic method estimates to projection selectivity are in many cases significantly lower than those by GP me-

thod. We observe that analytic method improves sensibly its performance if the projection is performed on two attributes respect to the case of three attributes. This result offers a relevant advantage when considering that plans of optimizers normally privilege selection and projection operations to the other more expensive operations. Therefore, the tentative of drastic reduction of query sizes using projection on few attributes can be supported by the availability of accurate estimates.



(a)



(b)

Fig. 4.1. Comparison of the performance of GP and CC methods for estimating projection selectivity respectively in the (a) bidimensional and (b) tridimensional cases

4.6 Conclusions

Determining the selectivity factor estimates of relational operations is a crucial task for optimizers to choose optimal paths of execution of query processing. The presented method adopts an analytic approach and stores all information of data profile by canonical coefficient series. It uses a unique method for representing both the multivariate data distribution and distinct value set. The analytic method, already successfully tested in the estimate of selectivity factor for restriction operation, has been adapted and tested here for estimating join and projection operations. The experimental results show good performance and improvements with respect to other conventional methods.

Moreover, its application is not limited to the estimation of selectivity factor of relational operations. Several multidimensional aggregate functions and statistical quantities can be easily and accurately estimated using canonical coefficients. This application has been receiving attention in emerging areas of database technology such as the approximate query processing field. It provides approximate answers to the queries very quickly and it is particularly attractive for large-scale and exploratory activities in OLAP applications.

References

1. Bell DA, Ling DHO, McClean S (1989) Pragmatic estimation of join sizes and attribute correlations. In: Proc IEEE 5th ICDE Conf., pp 76-84
2. Chaudhuri S and Narasayya VR (2001) Automating statistics management for query optimizers. IEEE TKDE 13:7-20
3. Chaudhuri S, Das G, Datar M, Motwani R, Narasayya V (2001) Overcoming limitations of sampling for aggregation queries. In: Proc IEEE ICDE Conf., pp 534-542
4. Chen CM and Roussopoulos N (1994) Adaptive selectivity estimation using query feedback. In: Proc ACM SIGMOD Conf., pp 161-172
5. Ciaccia P and Maio D (1995) Domains and active domains: what this distinction implies for the estimation of projection sizes in relational databases. IEEE TKDE 4:641-654
6. Gardy D and Puech C (1984) On the sizes of projections a generating functions approach. Information Systems 9:231-235
7. Gibbons PB (2001) Distinct sampling for highly-accurate answers to distinct values queries and event reports. In: Proc 27th VLDB Conf., pp 541-550
8. Haas PJ, Naughton JF, Seshadri S, Stokes L (1995) Sampling-based estimation of the number of distinct values of an attribute. In: Proc 21th VLDB Conf., pp 311-322

9. Ioannidis YE and Christodoulakis S (1993) Optimal histograms for limiting worst-case error propagation in the size of the join results. *ACM TODS* 18:709-748
10. Ioannidis YE and Poosala V (1995) Balancing histogram optimality and practicality for query result size estimation. In: *Proc ACM SIGMOD Conf.*, pp 233-244
11. Jiang Z, Luo C, Hou W-C, Yan F, Zhu Q (2006) Estimating aggregate join queries over data streams using discrete cosine transform. In: *Proc DEXA Conf.*, pp 182-192
12. Lefons E, Merico A, Tangorra F (1995) Analytical profile estimation in database systems. *Information Systems* 20:1-20
13. Ling Y and Sun W (1995) A comprehensive evaluation of sampling-based size estimation. In: *Proc IEEE 11th ICDE Conf.*, pp 532-539
14. Ling Y and Sun W (1999) A hybrid estimator for selectivity estimation. *IEEE TKDE* 11:338-354
15. Merrett TH and Otoo E (1979) Distribution models of relations. In: *Proc 5th VLDB Conf.*, pp 418-425
16. Mullin JK (1993) Estimating the size of a relational join. *Information Systems* 18:189-196
17. Sun W, Ling Y, Rishé N, Deng Y (1993) An instant and accurate size estimation method for joins and selection in retrieval-intensive environment. In: *Proc ACM SIGMOD Conf.*, pp 79-88
18. Vitter JS and Wang M (1999) Approximate computation of multidimensional aggregates of sparse data using wavelets. In: *Proc ACM SIGMOD Conf.*, pp 193-204

Chapter 5

Decision rules: a metamodel to organize information

Sabina-Cristiana Mihalache

Business Information Systems Department, Alexandru Ioan Cuza University of Iași, Carol I Blvd, no. 700505, Iași, ROMANIA, sabina.mihalache@gmail.com

Abstract. This paper try to introduce the idea that the form of specialized knowledge relating to decision models, rules, strategies may be a way to organize information by attaching meaning to the existing formalized entities at the computer-based information system level. We realize an exemplification by using a rule in order to attach a meaning to an existing class from the domain model in the form of a new property. The paper presents some technologies and tools that we used for the exemplified decision problem, some remarks and conclusions. We titled the article “thinking in decision rules” starting from the assumption that the way we think when we make decisions should be the metamodel to organize information. When we ask to obtain information we do not ask for specific data, but we ask on concepts that are usually interrelated in IF...THEN...ELSE rules.

Keywords. Decision model, metamodel, information retrieval, OWL, RDF, Protégé, SPARQL, ontology, inference engines, and computer-based decision models

5.1 Introduction

Considering decisions a matter of problem-solving transforms data and information models into problems' models. Information that is necessary in

making decisions is a metamodel to organize information in a way that proves to be useful for decision-making. Problems' models organize data that is necessary in evaluating decision alternatives and optimize the value for each alternatives conforming to a set of criteria. Information models organize data conforming to an accepted common sense for data in a way that maps reality through representing knowledge.

The literature concerning computer-based decision models and decision-problems search to provide general problem-solving models applicable in all sorts of situations. Either is the form of analytical models, data mining models, expert systems models they all embed a model for solving a problem and transforms from what intended to be general into a specifically solution. All of these models prove to be static, non-adaptive, and incapable to react when the conditions change or interact with a decision-maker. The idea to realize decision models is good and necessary, but the technology is not sufficient. We try to argue in this article that the whole idea of realizing decision models should be used in improving information provided in decision-making and all that is necessary is to create an environment, a metamodel to organize information so the decision-makers can specify the decision-rules that constraints a problem.

5.2 Problem formulation

The real problem in specifying decision-models consists in representing knowledge as control constructs. It is ordinary known that in order to make a decision there are necessary information and knowledge application. Information technologies and technology' providers have tried always either to offer possibilities to embed decision problems simulations either to provide a way to have easy access on aggregated data, usually named information. Decision support systems and Business Intelligence technology's providers have always tried to animate the idea that analyzing data means structuring problems and therefore information and knowledge is disseminated to the user. But knowledge in its pure sense is a pure form of what proved to be true, either changed conditions. What is true have always an unambiguous value and not a fuzzy value or a probable value derived from some comportment presented for data sets.

What we try to say is that the philosophy presented for some technology trends is good, but the infrastructure or the metamodel for decision modeling is not unified probably because there are many different approaches in analyzing decision problems. In the following we will present two problems: 1) the output for a decision-model: a result of an act or information

that can transform in action? 2) the ambiguous ideology of knowledge representation.

5.2.1 The output for a decision-model

Making decisions concretizes in the phase of implementation, in actually performing an action. We have a philosophy concerning decision-making steps provided by H.Simon which is unbeatable because is always true: we always make decisions by choosing from some alternatives one alternative that satisfies best a set of criteria preferences. Modeling decisions doesn't mean that it has to be realized in the same way: implementing the alternatives, the criteria and the algorithm that optimizes the criteria. We think in a specialized way so it can not be provided a general model for decisions by implementing models as alternatives and choosing on the respective alternatives. The output for a decision-model should have to be information with potential in activating an action and not the action itself. The information provided should refer to possible effects for a decision and not the actual way for acting because often the decision-maker will not trust the model: the respective model hides knowledge that is implemented in the form of control constructs. Usually the decision-maker doesn't know why the implemented model provided a solution because he/she and the model do not share the same meanings and amount of knowledge.

During the time we have experienced different forms of technology: from data processing (which is actually the "intelligent" form for computer-based decision models) to knowledge repository, data mining and knowledge discovery, knowledge-based systems or even the buzz form business intelligence.

It is not actually the fault of the technology; we consider that the fault belongs to the ideology in using these technologies. We should not believe that processing data means providing intelligent data or that if we deposit knowledge we actually do. It can not be the success of one technology because we had so many examples that failed in the past. All that ideology of decision-support systems failed because the implemented models didn't proved to be interactive: they had a major problem in interacting with user because humans and computers don't understand symbols in the same way. So the technology providers embedded all kind of data structures and representation formats in order to get more closed to the user. But, in the moment, the computer-based models and the user can not constitute both in a system, they can not be integrated.

The output for a decision model should be information capable to activate action. The respective action should be user's desire, wish, believe

and the information must refer to that. It doesn't make any sense to try to emulate all kind of desires, wishes, beliefs because is unattainable obviously. The implemented models must provide a way to the user to specify his/her beliefs or wishes and in that way the computer-based models be able to transmit actionable information.

5.2.2 The ideology of knowledge representation

The ideology of representing knowledge belongs to Artificial Intelligence and it had interesting approaches during the time. The knowledge level was firstly proposed by A. Newell who considered that in order to develop intelligent models there is a need to represent a knowledge level which is above the physical level. This idea conducted studies in developing inference engines but also provided a basis for data modeling. The knowledge representation principle consider that differentiating knowledge by structuring it conforming to different roles that pieces of knowledge may have could provide easier access to data.

Knowledge representation principle offered during the time different paradigms for programming languages: frame-based, object-oriented. Either be the paradigm its philosophy, tools, languages and data structures try to virtually connect data structures to knowledge from a reality known as discourse universe. At the physical level there is all about data structures and algorithms. Developing models transforms in implementing computer-based applications in a programming language. The Artificial Intelligence critics said always that implementing models by specifying all the existential conditions doesn't mean Artificial Intelligence: it is all about data structures, procedures and control structures.

On the other side, knowledge is pretty difficult to define. Knowledge is something that is always true and as Einstein said: the truth is what passes the test of experience. Representing the world and its existence also has connections in philosophy by the approaches concerning ontology.

The literature from knowledge-based systems technology field provided different approaches. The most pragmatic that we consider are inference engines and ontologies. All the principles and philosophy that other technologies promote are good but these two technologies are more oriented on knowledge and representing concepts. The expert systems technology, the fuzzy systems technology, neural networks technology are more oriented to problem solving. This aspect of problem solving conducted to considering only some sort of problems being satisfactory for Artificial Intelligence: diagnosis, monitoring, explanation. In fact what proves to be important in developing these models isn't the knowledge that they embed but

the algorithms used to manipulate that knowledge. All the know-how used on knowledge is embedded in a hidden form of procedural knowledge like: forward chaining and backward chaining, fuzzy algorithms, genetic algorithms, stored procedures.

So, in order to actually implement declarative knowledge we must have access to larger represented data sets, existential knowledge that always is true like ontologies. The technology to be used must have implemented general forms of inference on the concepts from ontologies. All the conditions, constraints, controls on concepts must be edited by the user, by editing the metamodel of inference.

So we can formulate another principle for knowledge representation: knowledge engineering must provide the metamodel used in inferring and instantiating knowledge. Instantiating knowledge from the values specified for control data structures must be realized by the users of the metamodel: inferring knowledge doesn't belong to knowledge engineering, but the metamodel does.

The specialized knowledge belongs to users and the user location represents the place where knowledge application is performed. General knowledge and its forms of theories, axioms, and ontology are very little supported by technologies and usually are confounded with specialized knowledge. Even a mathematical model starts from a set of assumptions in order for a problem to have a solution. The mathematical models aren't a form of general knowledge; they are a form of procedural knowledge.

We ask what is the metamodel for decision problems. Everyone makes decisions in terms of IF...THEN...ELSE rules. The metamodel of IF...THEN...ELSE rule must be implemented in technology and the metamodel editing must be realized by the user.

5.3 Problem solution

The main solution that we provide in assuring the generality for decision-models is considering the metamodel being one of importance and not the specific conditions that characterize a certain problem.

From the informatics point of view we propose the following technologies: relational databases (to organize and store data), ontology (to organize meanings), inference engines (to organize constraints, conditions, rules defined by the user).

Inference engines are a technology that uses production rules, in the meaning that they provide an environment to infer derived knowledge from facts. We imagined a scenario in which the user can edit the decision

rules used for inferring facts from a decisional problem or situation. He/She use an environment that permits editing, using and questioning the individuals represented in ontology. The individuals are the instances from a database or another source for data written in a format that an ontology editor may recognize.

Supposing that at one moment the user say: <<“IF property a for class A is greater than 2 and property b of class B is greater than 3 THEN C”; “IF C and property c of class D is greater than 5 THEN E”>>; the user must have functionalities provided by technology in order to specify this and actually infer the new facts (new properties or new classes). After doing so, maybe she/he wants to query the ontology which individuals respect his/her constraints. By implementing the metamodel without specifying the control on data, the answer provided will be semantically enriched if the rules edited by the user are fired on ontology’s individuals.

The decision model remains valid even if in the near future the constraints actually change and the user will may write his/her new rules of thinking without changing the metamodel.

From our experience in modeling decisions during the PhD program and two research grant programs that we manage we observed that all the literature related to decision modeling refers to analytical models usually based on mathematics or expert systems models based on the so called knowledge captured from experts. In fact, in practice these models are not used. Every idea, philosophy or technology proves its usefulness by using it in practice. But we can not say that the analytical models or expert systems models are used in practice because these models have an embedded form of knowledge hidden in some sort of technology. In our opinion, for the decision-makers use some models it is necessary that they can specify the constraints, the inferring chains of rules without knowing the intrinsic part of the metamodel.

The actual infrastructure to organize data and have access on information is presented in Figure 5.1.

The metamodel must be represented on the concepts and the metamodel editing must be performed by the user.

A basic question is why do we use the term concept? A concept is better suited to a class of objects that contains instances; it describes an entity someone would say. Not exactly: our opinion is that a concept is a combination of instances that characterizes different entities. And they should be different in order to provide a special meaning in the sense of a concept, otherwise is a class or a table or an entity.

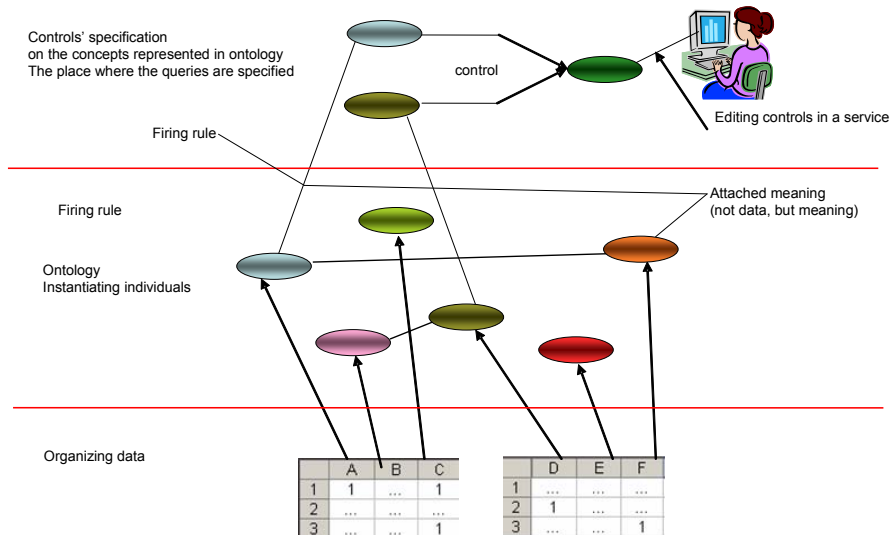


Fig. 5.1. The places where the control on concepts is specified

The expert system model technology represented a good solution for combining concepts in IF...THEN...ELSE rules which represented the key of this technology and the best suited way to represent the specialized knowledge provided by an expert. The major disadvantage was the fact that an instance from the database source corresponds to a single concept. The mapping between the concepts and data was at the instance level which implies a redundancy of concepts because the concepts aren't organized as data. Organizing concepts otherwise than in the form of IF...THEN...ELSE rules must provide a way to represent information, to attach meaning in a way that humans do. Humans recognize, classify and evaluate messages in order to perceive and to attach meaning. This would have to be a method to organize concepts to represent some sort of meaning and this kind of methods use semantic technologies models.

5.3.1 An example to justify the solution

In the following we will present a problem and a possible conceptualization to this problem to sustain our ideas.

In order to provide information there must be a mapping between different data sources stored usually in relational databases. Considering the case of a simple decision anyone can assume that the data sources are heterogeneous. If we take as example the case of deciding if an asset is impaired, data stored in the organization's database aren't sufficient. We

must take into account the market value for the asset or the so called capitalized value or utility value. Assuming that for finding this value someone would choose the Internet the data found for the respective value would have to be saved in a file format that other program should “understand” which for an application means read and perform a predefined function, calculus or execute a procedure on that data. For that data to be combined with the data derived from a database they must share a common ontology. Usually for data extracted from a database it is recommended the RDF [13] format which is a description in the form of triples, each consisting of a subject, a predicate and an object. We exemplified this problem by using an external table in PostgreSQL which stores data resulted from a query performed on a databases which contains more tables and by using an .xml file as we would save the data from a web site.

To transform the two different formats in which we have stored data about assets we used D2RQ [1] which is a declarative language to describe mappings between relational database schemata and OWL/RDFS ontologies. The mappings allow RDF applications to access the content of huge, non-RDF databases using Semantic Web query languages like SPARQL. The mapping is actually quite simple to make and implies two command lines in the MS-DOS prompt.

After that, we combined the two files simply by eliminating the redundant triples.

The standard language recommended to build ontology is OWL [14]. There are a lot of ontology editors but the most known tool is Protégé Editor which is open-source (there is a big community who uses this tool to develop ontologies and to build plugins). In order to transform the file from RDF format to the OWL format we used SWOOP [6] which is another open-source tool that supports the RDF format but it doesn't have so many plugins as Protégé has. We simply saved the file in the owl format.

Finally we were being able to see the database relational schema as ontology. We can observe that the ontology refers to a super class called **THING** which belongs to Protégé and that we have a single class whose name corresponds to table's name from relational database schema; the class has properties whose values are the fields' names and individuals whose names corresponds to table's attributes values. We can say in this moment that we managed to combine two different data sources that share the ontology defined by the table source data and that this data are the resulted stored in one table from a query addressed to an initial multiple-tables database which had to respect some integrity constraints.

In this moment it is good to ask ourselves if we have represented concepts. And a good answer is: not so much according to our above defini-

tion. We have data stored in individuals that have properties that belong to a class so we may say that the model could present the capacity to recognize and classify, but not to evaluate because the data represented has no attached meaning.

In order to evaluate this meaning for a computer W3C recommends SWRL Semantic Web Rule Language which permits editing the rules of existence for the data to be recognized in ontology. SWRL permits only editing the rules and not firing on them (capability that would permit adding a property value to an individual and in this way reaching the model's semantics).

Under the umbrella of the rule engines are a lot of tools but because Protégé is Java based the Java engine recommended is Jess. The users' community developed also a JessTab Plugin which is available with the last version of Protégé. Jess is a rule engine that derives from CLIPS and uses Horn clauses to represent the rules. The users' community also uses bridges to map SWRL Rules into Jess rules but because this is not a trivial task we managed to do it without using a bridge. So we wanted to attach the meaning that for an asset to have the capitalized value under the net accounting value is to be impaired.

Once the relevant OWL concepts and SWRL rule have been represented in Jess, the Jess execution engine can perform inference. As rules fire, new Jess facts are inserted into the fact base. Those facts are then used in further inference. When the inference process completes, these facts can then be transformed into OWL knowledge, a process that is the inverse of the mapping mechanism.

After firing the presented rule the facts stored in Jess are the same but they have an additional slot named "impairment" if the condition specified in the rule is true. In our case, only one fact presents this aspect and we present the fact in Figure 5.2. There exists a slot named "depreciere" (impairment) that we defined in OWL ontology as a property of `vocab0:mijlocfix_nrinventar` (fixed_asset) with the accepted values "da" (yes) and "nu" (no). The slot does not belong to the ontology provided by relational database schemata, it was defined by us and its value is attached to the individuals only if the rule proves to be true.



Fig. 5.2. The fact that contains the additional slot “depreciere” (impairment)

What we deliberately omitted to say is that we used the D2RQ tool especially because it provides a way to visualize the individuals from ontology in a browser, so in order for a user to extract useful information it is

sufficient to access the n3 format of the ontology in a browser. D2R Server address is <http://localhost/2020> and it permits viewing data in n3 and in the RDF format and more important querying the ontology by using SPARQL. Protégé doesn't permits saving the owl format back into n3 format once the OWL ontology has individuals defined by using rules.

5.4 Conclusions

Better decisions means improving the information provided. We tried in this paper to outline that the knowledge of acting from the decision models must be implemented to improve information, to actually provide better information to the user. Using the decision models for problem-solving task proved to be a failure in the past. It concluded in very static models, non-adaptive ones, with no utility for the user because they captured a kind of model that impose performing an action by the decision maker in the form of transferring knowledge from the model to human being.

We do not try to argue that this is the best model that we want to recommend but in the moment the technology and the technology's producers aren't too concept-oriented in the business domain although there is a clear wish to solve the problem of interoperability. Only for our little example we used no more than six tools to manage to transmit a simple message and we are not saying that this is a sure way to represent meaning. The tools that we used have a lot of facilities that we maybe omitted to consider.

We provided this model oriented somehow on decisions due to the last standardization efforts in the field of information integration and due to the fact that we study decisions models as a special concerning in this moment.

We considered all the possible applications in practice: decision-support systems (this is a field where information is considered to be some aggregated data and not meanings organized in ontologies), information retrieval, knowledge engineering, knowledge-based systems project and management.

References

1. Bizer C (2006) D2RQ V0.5 - Treating Non-RDF Databases as Virtual RDF Graphs. At <http://sites.wiwiiss.fu-berlin.de/suhl/bizer/D2RQ/>, accessed 30 June 2007

2. Chandrasekaran B (1990) Design problem solving: A task analysis. *American Association for Artificial Intelligence* 11(4):59-71, <http://www.cse.ohio-state.edu/~chandra/ai-mag-design-ps.pdf>
3. Chandrasekaran B (1988) From numbers to symbols to knowledge structures: AI perspectives on the classification task. *IEEE* 18(3):415-425, <http://www.cse.ohio-state.edu/~chandra/From-numbers-symbols-knowledge.pdf>
4. Chen H, et al. (2006) Toward a semantic web of relational databases: a practical semantic toolkit and an in-use case from traditional chinese medicine. In: ISWC 2006 Conference, <http://iswc2006.semanticweb.org/items/Chen2006kx.pdf>
5. Fayyad U, Shapiro G, Smyth P (1996) From data mining to knowledge discovery in databases. *American Association for Artificial Intelligence*, www.aaai.org/AITopics/assets/PDF/AIMag17-03-2-article.pdf
6. Google (2007) Semantic Web Ontology Editor. At <http://code.google.com/p/swoop/> accessed 30 June 2007
7. Musen M (1998) Modern architectures for intelligent systems: reusable ontologies and problem-solving methods. In: *AMIA Annual Symposium*, Orlando, FL, <http://smi.stanford.edu/smi-web/reports/SMI-98-0734.pdf>
8. Newell A, Simon H (1963) GPS, A program that simulates human thought. In: Feigenbaum EA, Feldman J (eds) *Computers and thought*. (1995) *American Association for Artificial Intelligence Press Edition*, pp 279-293, <http://www.cog.jhu.edu/faculty/smolensky/050.326-626/Foundations%20Readings%20PDFs/Newell&Simon-1963-GPS.pdf>
9. Protégé (2007) The Protégé ontology editor and knowledge acquisition system. Stanford University, at <http://protege.stanford.edu/>, accessed 30 June 2007
10. Schreiber G, et al. (2000) Knowledge engineering and management – The CommonKADS Methodology. MIT Press, London, England, pp 403-418
11. Studer R, Benjamins VR, Fensel D (1998) Knowledge Engineering: Principles and Methods. *Data & Knowledge Engineering* 25(1-2):161-197 <http://hcs.science.uva.nl/usr/richard/postscripts/dke.ps>
12. Tang Z, MacLennan J (2005) *Data mining with SQL server 2005*. Wiley Publishing Inc, Indianapolis, Indiana
13. W3C, Resource Description Framework (RDF). At <http://www.w3.org/RDF/> accessed 30 June 2007
14. W3C, Web Ontology Language (OWL). At <http://www.w3.org/TR/owl-features/> accessed 30 June 2007

Chapter 6

A computerized solution for the financial diagnose of the SMEs

Vasile Daniel Păvăloaia

Business Information Systems Department, Alexandru Ioan Cuza University of Iasi, Bdul Carol I nr.22, Iasi 700505, ROMANIA, danpav@uaic.ro

Abstract. In the now-a-days business environment, when efficiency and profitability are two major aspects to be taken into account by any enterprise, including the small and medium sized enterprises (SMEs), interested in obtaining better economic results, the use of computerized tools for performing the financial diagnose on the activity of an enterprise is a must. From the multitude of information technologies that can be used in order to design such a tool, we choose Exsys Corvid as this software product is very reliable to use, allows integration with other programs (database query) and has its own browser to run the application. One of the papers' major aims is to illustrate a manner of computerizing the field of financial diagnosis for the case of a Romanian enterprise. Thus, the current study will be conducted by implementing into a real application the theory (from the specialized literature) and expertise (collected from experts) related to the field of financial diagnose and will utilize, for exemplification purposes, the financial ratios that are used in the process of a financial diagnose. The computerized solution can be easily implemented for the case of the SMEs as well as for any other enterprise as long as the bankruptcy prediction model used is the proper one.

Keywords. Financial diagnose, Intelligent systems in Economy, Computerized solution, Economic and financial analysis, Exsys Corvid

6.1 Introduction

The financial phenomenons are characterized as being restless and they modify it structure under the impulse of the factors that generate the dynamics of social life. In the light of this information, a system based of financial ratios should be extremely flexible and sensitive in order to appropriately observe and reflect the reality. As technology progresses rapidly, there must be used specific tools for managing such a system that is made up of financial ratios, in order to speed up the time consumed by the ratios commensuration. At the same time, it is required to improve the quality of decision in the field of financial diagnose [1]. This task is only accomplished by human experts that acquire knowledge throughout many years of practice. Therefore, we intend to capture such knowledge and implemented into the computerized solution.

From a theoretical point of view, the causes and factors that lead an economic entity to bankruptcy are not only of financial nature and they can conduct to a low profitableness and solvency of the enterprise. The main information obtained from the literature, related to the field of financial diagnose, is that an enterprise will encounter major difficulties in fulfilling its relationship with the contractor's if it doesn't have a minimal rate of profitableness and a satisfactory level of liquidity.

Therefore, in order to ensure its survival (if not the growth), the enterprise must conserve its financial autonomy; otherwise, the investments that require supplemental financial resources cannot be fulfilled. The lack of profitableness implies an insufficient level of auto-finance capacity with respect to the required investment rate. On the other side, the lack of profitableness obliges the enterprise to become indebted as a low level of profitableness doesn't allow it to attract new capital from investors. This measure will lead to an increase of the enterprise expenditures, affecting its solvency and furthermore, leading to incapacity in terms of payment.

Ratios values		Profitableness	
		positive	negative
Liquidity	positive	1 „in shape”	3 „chronicle malady”
	negative	2 „transient malady”	4 „imminent ending”

Fig. 6.1. Profitableness, liquidity and the financial health of the enterprise

As a result, one may notice the importance of maintaining an economic equilibrium for the enterprise functionality which, mainly, resides in performing a profitable activity concurrent with monetary equilibrium – all of

those actions are translated into a satisfactory level of liquidity. Having in mind the importance of profitability -liquidity correlation required for the persistence of an enterprise, the specialized literature value the “health state of an enterprise” by including them into four groups, illustrated in Figure 6.1.

As an interpretation of the information included in Figure 6.1, the enterprise that suffers from „transient malady” has a certain level of profitability but encounters some liquidity problems. It is the case of first-years enterprises, in full growth, whose investments in capital assets and working capital are higher than its auto-finance capacity. These economic entities are less exposed to the risk of bankruptcy if they can overcome the liquidity issues.

Whether they are young or mature, the enterprises that belong to the third category „chronicle malady” have profitability issues related to either the difficulties in maintaining the market share or the high weight of fixed expenses within the quantum of the total manufacturing expenses. When the total collected income covers the payment expenses and the investments are kept within a normal level, the liquidity ratio can be rated as satisfactory, allowing the enterprise to develop a profitability activity.

An economic entity that is included in the fourth category „imminent ending” it cumulates an insufficient profitability and a total lack of liquidity ratio. Its survival can only be attained after profound reorganization, accompanied by a reinforcement of the incoming financial resources.

As a consequence, in the Romanian economic system, in order to perform an enterprise diagnose, it is required to test the liquidity rate and profitability of the economic entity. In the following chapters of the paper we will try to formulate and solve this matter.

6.2 Problem formulation

As stated in earlier paragraphs of the paper, in order to perform an enterprise diagnose, it is required to test the liquidity rate and profitability of the economic entity.

6.2.1 Testing the enterprise liquidity ratio

In order to test the enterprise liquidity level there it is used a ratio system illustrated in Table 6.1; the best rate of the liquidity ratios will be attained when an enterprise fulfills all the conditions imposed by the ratios (Table 6.2).

Table 6.1. The system of Liquidity ratios

Ratios name	Formulas	Explanations
General liquidity rate (RLG)	$=Ac/Dts$	Ac = Current assets; Dts = Short term liabilities;
Low liquidity ratio (RLR)	$=(Ac-St)/Dts$	St=Reserves;
Immediate liquidity ratio(RLI)	$=Disp/Dts$	Disp = Liquidness;
The rate of covering the pay- ment engagements (Ro)	$= FTE/ (A+ Div_{plătite}+Inv)$	FTE = Cash flows; A = Annuity; Div _{plătite} = Paid dividends; Inv = Investment payments;

Table 6.2. The interpretation of the ratios values

Liquidity ratios	Obtained values	
	Favourable	Unfavourable
RLG	$RLG \geq 1$	$RLG < 1$
RLR	$RLR \geq 0.8$	$RLR < 0.8$
RLI	$RLI \geq 0.2$	$RLI < 0.2$
Ro	$Ro \geq 1$	$Ro < 1$

6.2.2 Testing the enterprise's profitability

Enterprise's economic and financial performances can be measured throughout N note that is obtained by averaging five ratios and the value of each of ratio is determined by using the following pattern (see Eq. 6.1):

$$\frac{\text{Value of } R_i \text{ ratio}}{\text{Default value of } R_i \text{ ratio}} \quad (6.1)$$

The default value of R_i ratio is determined for a population obtained from the default enterprise and turns out to be the median of the enterprises that belongs to the same sector of activity. The general model of a score function that gives a mark for each particular case, is as follows (see Eq. 6.2):

$$N = a_1R_1 + a_2R_2 + a_3R_3 + a_4R_4 + a_5R_5 \quad (6.2)$$

The enterprise that has a perfect state according with a default state has a mark N computed as follows (see Eq. 6.3):

$$N = a_1 + a_2 + a_3 + a_4 + a_5 \quad (6.3)$$

Building and applying such a model may look like a simple task but its relevance was acknowledged by the empiric tests which were conducted in order to verify the statistic method.

$$Z = 0.24X_1 + 0.22X_2 + 0.16X_3 - 0.87X_4 - 0.10X_5 \quad (6.4)$$

The Conan-Holder model had been statistically tested and it is associated to a bankruptcy probability in a score function [2]. The functions suggested for the industrial (see Eq. 6.4) and manufacturing (see Eq. 6.5) sectors have the following pattern:

$$Z = 0.0197X_3 + 0.0136X_2 + 0.0341X_6 + 0.0185X_7 - 0.0158X_8 - 0.01222 \quad (6.5)$$

The purports [3] of the parameters used in the above models are explained in Table 6.3.

Table 6.3. The parameters used in the Conan-Holder model

Parameter	Formulas	Explanations
X_1	=EBE/Dt	EBE = Gross operating excess; Dt =
X_2	=Cperm/At	Total liabilities; Cperm = Permanent
X_3	=Disp/At	capital; At = Total asset; Disp =
X_4	=Chfin/CA	Money liquidity Chfin = Financial
X_5	=Chpers/Vad	expense; CA = Turnover (except
X_6	=Autofin/Bt	VAT); Chpers = Employment ex-
X_7	=EBE/Bt	penses; Vad = Added value; Autofin
X_8	=NFR/CA	= Auto-financing; Bt = Total Balance
		sheet; NFR = Operating capital

The obtained score value can be explained by using the interpretations showed in Table 6.4.

Table 6.4. Score functions value interpretation for different activity sectors

Value interpretation	Manufacture	Commerce
Good shape	$Z \geq 4$	$Z \geq 20$
Danger	$Z < 4$	$Z < 20$

In order to test enterprise's liquidity and profitableness ratios, the input data will be taken over from a spreadsheet file that has the structure shown in Figure 6.2.

	A	B	C	D	E	F	G
4							
5	Parameter	Ac	Dts	St	FTE	A	Divp
6	Value	73492284	79840528	40522862	100000	20000	132450
7							
8	Parameter	Inv	Disp	Rec	At	Rnet	Cpr
9	Value	500000	6166480	17057763	1064528882	5213587	25669712
10							
11	Parameter	d	CA	EBE	Dt	CP	Chfin
12	Value	10,5	172907428	9050306	942642	26612354	10851112
13							
14	Parameter	Chpers	Vad	AF	Bt	NFR	
15	Value	10995878	20466383	12762482	2129057764	73492284	

Fig. 6.2. The spreadsheet file containing the parameter's values

6.2.3 Inferring the recommendations

If the enterprise is diagnosed as “in shape” the user of the computerized solution will be prompted to choose whether he wants to find out about the recommendations and solutions he has in order to maintain and improve the enterprise performances. In this case, the ratios whose values are near to the normal limit are tested in order to identify the actions that can be undertaken so that the ratios don't run over their normal limits.

When the enterprise is diagnose with any of the other states (“transient malady”, “chronicle malady” or “imminent ending”) the situation is consider to be critical. In this case, the user will be prompted to find out the causes that led to this dangerous situation. This action allows the analyst to identify the threats and weaknesses that generated the abnormal functionality of the enterprise.

6.3 Problem solution

Exsys Corvid provides a powerful environment for developing Web-enabled Knowledge Automation Systems for a wide range of decision-making problems. This software allows the logical rules and procedural steps used by an expert to reach a decision, to be efficiently described in a manner that is easy to read, understand and maintain. Corvid converts logic to a form that the underlying Runtime Inference Engine can process in order to emulate the questions, process, and recommendations of the expert in an interactive session, allowing the end users to interact with the system as if they were talking to the expert, to produce situation-specific recommendations and advice on a wide range of subjects [5].

The computerized solution, implemented with Exsys Corvid, took advantage of all the above features of the software that, throughout its rule base, queries an external spreadsheet file by using a friendly interface. Thus, the main aim of the solution is to help the financial analyst run a financial diagnose on an enterprise only knowing the values of certain financial parameters (shown in Figure 6.1). The computerized solution has two goals:

1. to establish, based on the input information, the financial state of the enterprise (one of the four category illustrated in Figure 6.1);
2. to reveal the causes that generated the particular financial state of enterprise.

All the information included in Tables 6.1 to 6.5 was implemented into Corvid under the form of variables, logic and command blocks, all of them forming the rule base of the system.

6.3.1 The system's variables

All the elements that are used in the formulas presented in Tables 6.1 and 6.2 were implemented as Corvid numeric type variables. Their values were obtained by querying the spreadsheet file shown in Figure 6.2. From the specialized literature and also from the domain expertise, all the possible recommendations were implemented as Corvid confidence type variables. Some information was required from the user and thus, several variables were inputted as static list type variables. The system has a total of sixty four variables of the three types mentioned above.

When working with numeric type variables, for best results, it is required an integration of the expert system with an external database that contains the elements that the ratios system uses. This action can be best accomplished by using a spreadsheet file as an external database. Among the reasons that brought us to the above statement we only stress out two of them: the ease of working with this type of database and also the availability of the spreadsheet programs among computer users. In this manner, the "raw" data provided by the spreadsheet file would be processed by expert systems prototype.

6.3.2 Logic blocks

The reasoning base of the system is formed, in Corvid, by the logic blocks. A logic block will contain a certain combination of variables and it must obey the IF-THEN-ELSE reasoning control. For instance, in Figure 6.3,

we illustrate the logic block that contains the reasoning attached to the profitability ratio specific to the commerce activity. On the first line it is defined that the logic block applies only when the sector of activity is commerce; on the second line the formula is implemented – the values for the parameters are taken from the external spreadsheet file; the third line contains the recommendation: good level of profitability. The fourth line contains the specifications for the case when the profitability has an inadequate value (line five).

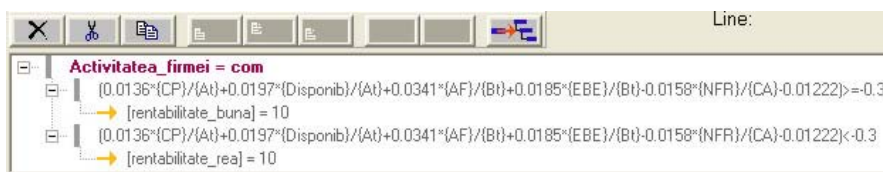


Fig. 6.3. The logic block containing the profitability ratio specific to the commerce sector

In the same manner were implemented all the required blocks containing the reasoning system. Another logic block, very important, that contains the reasoning for all the Liquidity ratios (illustrated in Table 1) can be viewed in Figure 6.4.

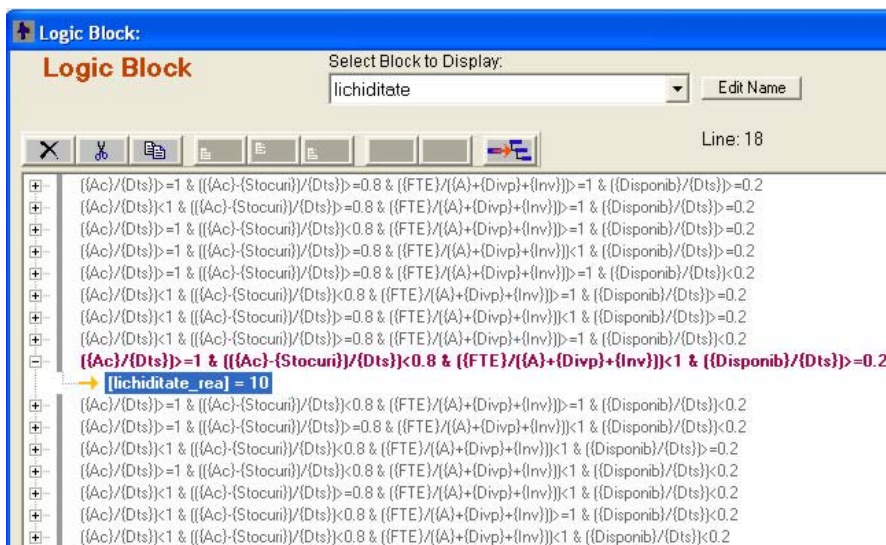


Fig. 6.4. The Liquidity ratios logic block

6.3.3 Command blocks

The Corvid inference engine runs the logic blocks in the ordered specified in the command block and infer the recommendations, namely the financial diagnose. Corvid allows both the backward and forward chaining, when it comes to setting up the inference strategy.

For exemplification reason, in Figure 6.5, it is illustrated the main command block of the system that sets up the strategy applied for the functionality of the system.

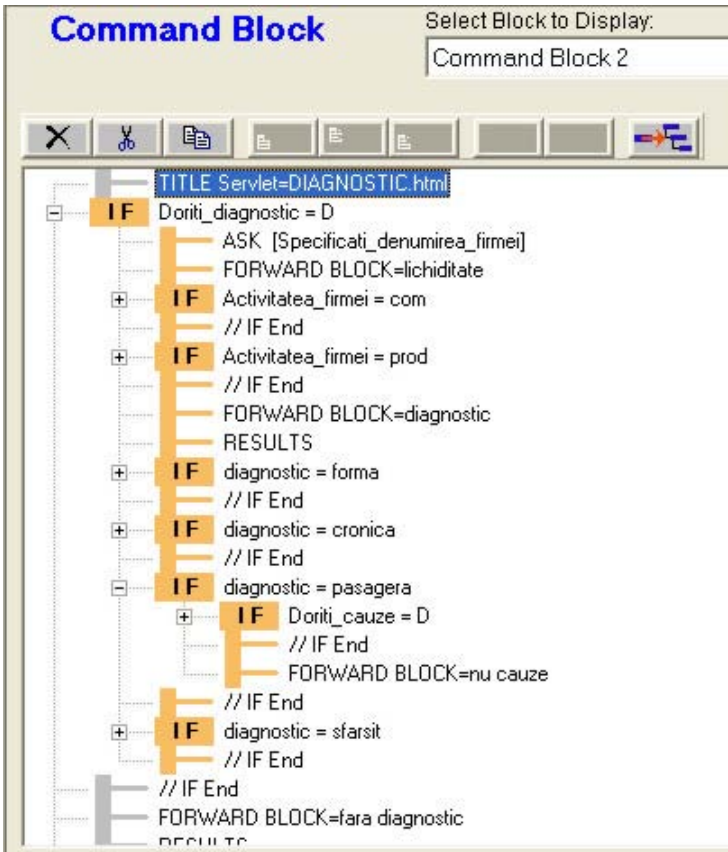


Fig. 6.5. The main command block of the system

3.4 Testing the computerized solution and displaying the results

Once launched, the system displays the main screen (Figure 6.6) containing the name of the system and the OK button.



Fig. 6.6. The main screen of the system

The action of pressing the OK button allows the computerized solution to commence the financial investigations that will end up with displaying the financial state of the enterprise (Figure 6.7).

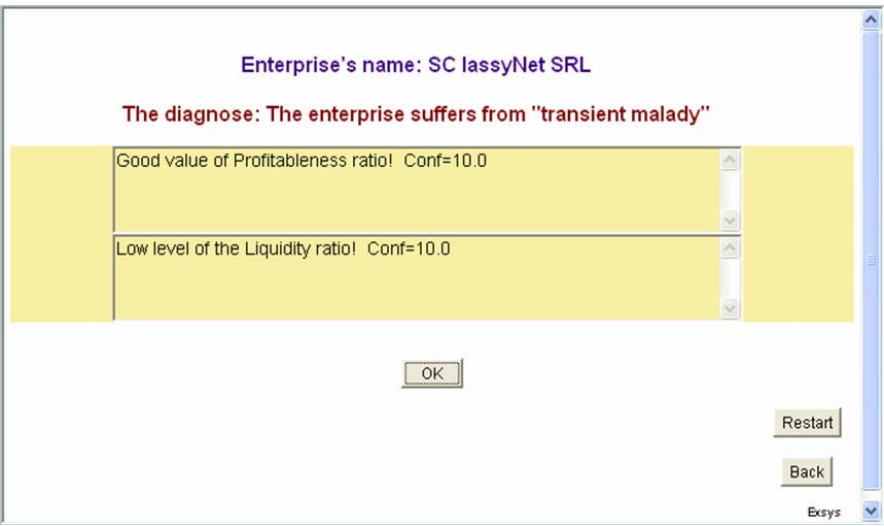


Fig. 6.7. The screen containing the financial state of the diagnosed enterprise

Also, based on user's choice, the system will display both the diagnose and the causes that led to the current financial state of the enterprise (Figure 6.8).

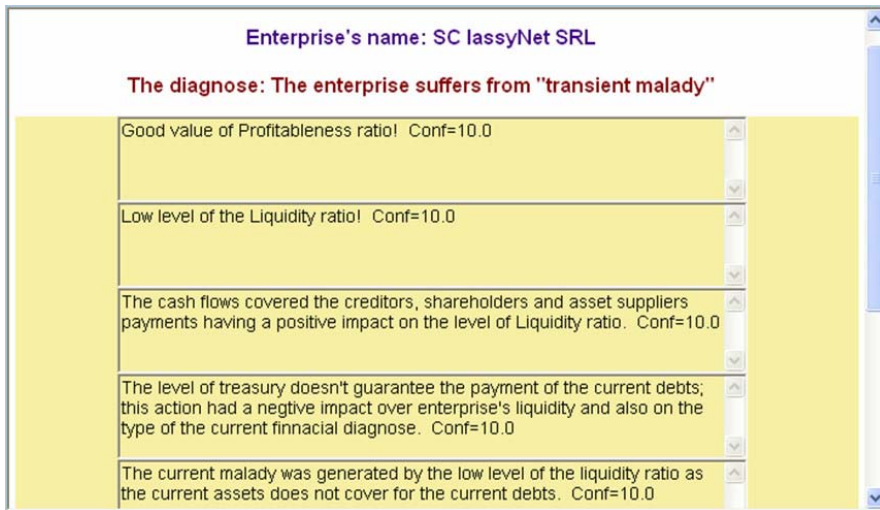


Fig. 6.8. The results screen containing both the financial diagnose and the causes

6.4 Conclusion

To conclude, we may say that Corvid is a competitive Expert System Shell that incorporates up-to-date technology and allows the developer to design and implement sophisticated expert system solutions for a variety of domains, including the economic field.

With powerful integration abilities (external data access), Java delivery through Applets and Servlets and strong interface design, Corvid stands for a leading software when it comes to developing prototypes for local or web environment. Therefore, one of the future directions for the development of the current research would be to incorporate it into a web page and make it available throughout the Internet. In this way, the current solution could be widely use by a variety of firms to test their financial performances.

The field of financial analysis and diagnose can be very easily computerized with the Expert system technology due to its abilities to interpret, analyze and infer recommendation and not just determine the value for a certain ratio.

References

1. Andone I, Pavaloaia VD (2004) Modelarea cunoasterii in organizatii. Metodologie obiectuală pentru soluții inteligente, Ed. Tehnopress, Iasi
2. Conan Holder (1979) Variables explicatives de performances et controle de gestion dans les P.M.I.. Universite Paris Dauphine
3. Pavaloaia W, Pavaloaia VD (2005) Analiza economico-financiara. Ed.Moldavia, Bacău
4. Roman M (2005) Statistica financiar-bancara. Editura ASE, pp 23-24, www.ase.ro/biblioteca
5. <http://www.exsys.com/productCORVID1.html>

Chapter 7

DELP system: Tracking deadlocks and phantoms in databaseas

Cristea Boboila,¹ Simona Boboila,² Constantin Lupsoiu¹

¹ Faculty of Mathematics and Computer Science, University of Craiova, Romania, {boboila, clupsoiu}@central.ucv.ro

² Computer Science Department, Stony Brook University, United States, mboboila@cs.sunysb.edu

Abstract. Database systems are the core of applications from various fields. In many of these, the well-functioning of databases can be extremely important, even critical. Deadlocks and phantoms are some of the problems that may appear in database systems, leading to information loss, with a highly detrimental impact. This paper presents DELP, a tracking system used to study different database access scenarios in which deadlocks and phantoms may appear.

Keywords. Database management systems, Concurrent transactions, Deadlocks, Phantoms, Locking.

7.1 Introduction

A database transaction represents a sequence of operations executed as a whole resulting in an interaction with a database management system. Considered independently from other transactions, a transaction should demonstrate coherence and reliability in all situations. On the other hand, when several transactions are running simultaneously, issues like deadlocks [1, 5, 6, 7] or phantoms [8, 9] are likely to appear.

In databases, deadlock refers to the situation when two processes are waiting to access a record, but they are blocked by each other; each of them waits for the other to release the lock on some record. The phantom phenomenon may occur when one transaction establishes a set of records as the result of a query and other concurrent transactions insert or delete records which would have been part of that set [11].

This paper presents the DELP (**DEadLocks** and **Phantoms**) system, a tool for analyzing how deadlocks and phantoms appear in a database management system. We present the system anatomy as well as the experimental procedure, and give a comprehensive description of current experimental results.

The rest of the paper is structured as follows: Sect. 7.1 presents related work in the area. Sect. 7.2 describes the system's structure and implementation aspects. Sect. 7.3 synthesizes the results obtained while carrying out experiments at different isolation levels. Sect. 7.4 concludes the paper.

7.2 Related work

Database management systems have benefited from extended attention in current literature. Our work adds to the research effort of tracing and preventing integrity constraints violations in transactional systems [2, 10, 13]. Moreover, several studies enlarge upon the effect of queries and updates during the interaction with the database system [3, 4, 14].

Deadlocks are among the main causes that prevent a correct functioning of systems in which several entities have to act concurrently. In a transactional system, simultaneous actions coming from many users may lead to deadlocks. A constant interest in recent research has been given to methods of database functionality optimization by preventing deadlocks [1, 5, 6, 7], as well as to methods of accurate informational data gathering by eliminating the risk of phantom occurrences [8, 9].

In contrast with most of current research, where theoretical surveys are presented, our paper targets a more practical perspective, and describes an environment that uses actual database technologies to trace deadlocks and phantoms.

7.3 System description

The DELP framework and the interaction flow between components are presented in Figure 7.1. The *Caller Module* communicates with the *DB Access Module* by invoking the procedure for database access.

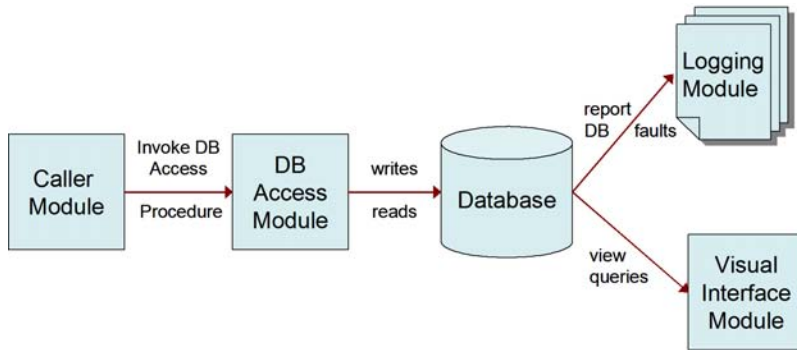


Fig. 7.1. System anatomy

Next, the *DB Access Module* directly interacts with the database, performing writes (updating and inserting) or reads (selecting). The *Visual Interface Module* delivers the state of the database to the user, upon user interrogation. Furthermore, the *Logging Module* records a series of statistics about the number of deadlocks, lost updates and phantoms that have occurred during the interaction with the database.

The *Caller Module* has been designed as a Java Client, which ensures that our implementation is cross-platform and provides a flexible database access mechanism. The *Caller Module* invokes the *DB Access Module* by directly calling the access procedure to the database.

The *DB Access Module* runs the SQL scripts which perform reads or writes on the database. The SQL scripts may be provided by the user as an input, or may reside in the shared space of the DELP system. In order to intercept the Java Client's requests, the database management system runs the Sybase server [12].

7.4 Experiments and results

We have run a series of experiments on a prototype database storing information about an on-line store. The e-store database contains prices and on-stock quantity associated with each item as well as other data about the products.

During our experiments, we have focused on how the state of the database changes when several transactions attempt to update and query it simultaneously. The DB Access Module runs the SQL scripts that interact with the database. For efficiency purposes, we have implemented the SQL scripts as stored procedures which reside on the server side (the Sybase server).

In order to achieve transaction consistency, it is essential what lock-based concurrency control mechanism is used by the database management system. At **SERIALIZABLE** isolation level, the transactions are isolated, and occur as if they were running one after the other. This way, the locking mechanism prevents consistency violations from happening. Due to the locking control, transactions may block each other and end in deadlock. One of the concurrent transactions must be aborted and rolled back. Our experiments in Sect. 7.4.1 study an illustrative scenario of deadlock occurrence in a database management system.

The next isolation level that we focus on is **REPEATABLE READS**, where the read locks are placed only on the records returned by a **SELECT** statement, and which are present in the database at a specific time. The read locks are not on the whole range requested with the **SELECT** statement, thus allowing new records that match the **WHERE** condition to be inserted in the database and generate phantoms (see Sect. 7.4.2).

In order to analyze the behavior of concurrent transactions, we have run multiple transactions in parallel during our experiments. The following results are obtained with 20 transactions running concurrently in our testing environment.

7.4.1 Deadlock occurrences

The stored procedure updates the price of one of the products in the e-store. Each manufacturer or brand has a specific policy of price update. For example, the manufacturer may decide to increase the price of its products when the number of items on stock falls below a certain threshold.

Therefore, our stored procedure must first count the number of items from the e-store with a read operation. Next, the same stored procedure modifies the price according to the policy. The following example presents the update procedure for a pottery manufacturer:

```
CREATE PROC UpdatePrice AS
DECLARE @count FLOAT
BEGIN TRANSACTION
SELECT @count =
(SELECT COUNT(*) FROM Products
```

```
WHERE Category = 'pottery')
IF @count < 1000 THEN
UPDATE Products SET price = 1.05*price
WHERE Category = 'pottery'
END IF
COMMIT TRANSACTION
```

The update action (database write), taken alone, poses no problems to the database management system. In the case presented previously, the write is preceded by a read (the select operation), and therefore the time interval that may interpose between select and update can be a source of deadlocks.

Since we have set the isolation level to be **SERIALIZABLE**, any of the 20 transactions can obtain the read locks. The conflict appears when one of the transactions wants to get the write lock to do the update. Neither of the transactions can obtain the write lock, because other transactions already hold read locks on the same database records, due to the **COUNT** statement. Since read locks are long at this level of isolation, none of the transactions will release the read lock until commit. Therefore, a deadlock appears and one of the transactions will eventually abort and give up the read lock.

7.4.2 Phantom occurrences

The next experiment studies database faults that may appear at **REPEATABLE READS** isolation level. The DB Access Module runs two different stored procedures that are executed independently from concurrent transactions. One of the stored procedures performs the price update:

```
CREATE PROC SetPrice AS
BEGIN TRANSACTION
UPDATE Products SET price = 1.05*price
WHERE Category = 'pottery'
COMMIT TRANSACTION
```

The second stored procedure carries out successive reads on the same data, in order to observe whether phantoms have appeared:

```
CREATE PROC ReadPrice AS
BEGIN TRANSACTION
SELECT price FROM Products WHERE Category = 'pottery'
SELECT price FROM Products WHERE Category = 'pottery'
COMMIT TRANSACTION
```

In this case, we observe that our database is prone to phantoms, because the update can take place in between the database reads.

Table 7.1 is a synthesis of the number of deadlocks and phantoms that occurred in the system when 20 transactions were running concurrently at the two isolation levels discussed.

Table 7.1. Deadlocks and phantoms

Procedure Name	Isolation Level	Fault Type	Number of Faults
UpdatePrice	SERIALIZABLE	Deadlock	34
SetPrice/ReadPrice	REPEATABLE READS	Phantom	15

7.5 Conclusions

The large applicability of database management systems in several fields that deal with information storage and processing has favored the perception of databases as one of the most important and demanding topics in today's applications and research. In this context, the well-functioning of the database system is imperative for accurate data.

This paper presents DELP system, a monitoring tool for database performance in different scenarios where concurrent transactions are accessing the database. Using DELP, the user may test the transaction in an experimental environment, and determine whether their interaction is detrimental, before running them in the actual storing environment. We are presenting a comprehensive experimental study, which points out the importance of choosing wisely the isolation levels for SQL databases. A thorough analysis of the situations in which deadlocks and phantoms appear can lead to significant better results in terms of information management in database systems. The current work focuses on tracing deadlocks and phantoms in the database management system. Future extensions include methods to automatically detect types of faults that may appear at other SQL isolation levels (e.g. lost updates, etc.).

References

1. Chen X, Davare A, Hsieh H, Sangiovanni-Vincentelli A, Watanabe Y (2005) Simulation based deadlock analysis for system level designs. In: 42nd Design Automation Conference

2. Decker H (2002) Translating advanced integrity checking technology to SQL. Database integrity: challenges and solutions. Idea Group Publishing, Hershey, PA
3. Elkan C (1990) Independence of logic database queries and update. In: 9th Symposium on Principles of Database Systems, ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems, pp 154-160
4. Godfrey P, Gryz J, Zuzarte C (2001) Exploiting constraint-like data characterizations in query optimization. In: ACM SIGMOD International Conference on Management of Data, pp 582-592
5. Kaveh N, Emmerich W (2001) Deadlock detection in distributed object systems. In: Joint Proc. of the 8th European Software Engineering Conference and the 9th ACM SIGSOFT Symposium on the Foundations of Software Engineering, Vienna, Austria, pp 44-51
6. Kobayashi N (2000) Type systems for concurrent processes: From deadlock-freedom to livelock-freedom, time-boundedness. In: Springer, Berlin Heidelberg, vol 1872/2000, p 365
7. Krivokapic N, Kemper A, Gudes E (1999) Deadlock detection in distributed database systems: A New Algorithm and a Comparative Performance Analysis. VLDB Journal 8(2):79-100
8. Rakow T, Gu J, Neuhold E (1990) Serializability in object - oriented database systems. In: 6th International Conference on Data Engineering, pp 112-120
9. Reimer M (1983) Solving the phantom problem by predicative optimistic concurrency control. In: 9th International Conference on Very Large Data Bases, pp 81-88
10. Ross KA, Srivastava D, Sudarshan S (1996) Materialized view maintenance and integrity constraint checking: trading space for time. In: ACM SIGMOD International Conference on Management of Data, pp 447-458
11. Silberschatz A, Korth HF, Sudarshan S (2005) Database system concepts, 5th edn. McGraw-Hill, Boston, MA
12. Sybase SQL Server™ Reference Manual. <http://download.sbase.com/pdfdocs/srg1100e/sqlref.pdf>. Accessed: March 15th, 2007
13. Vielle L, Bayer P, Küchenhoff V, Lefebvre A (1999) Checking integrity and materializing views by update propagation in the EKS system. In: Materialized views: techniques, implementations, and applications, MIT Press, Cambridge, MA
14. Wetzel G, Toni F (1998) Semantic query optimization through abduction and constraint handling. In: 3rd International Conference on Flexible Query Answering Systems, pp 366-381

Chapter 8

Concurrency control for multilevel secure distributed real-time databases

Valuchandhar Vadivelu,* S. Albert Rabara

*Department of Computer Science, St.Joseph's College (Autonomous),
Tiruchirappalli – 620 002. Tamilnadu, India.

Abstract. The technology evaluation of database management system has integrated with significant improvement in distributed computing. The requirements for multilevel security have been shown to conflict with database integrity and consistency. Concurrency control ensures that individual users meet consistent states of the databases with security. Concurrency control is an integral part of the database systems to manage the concurrent execution of operations by different transactions in the same data or distributed data with consistency. In this paper, it is proposed a model to enhance the performance of concurrent transactions for multilevel security for distributed database. This model reduces the data access time and wait time for every transaction monitored by the sub-query analyzer. Simulation study reveals that the effective enhancement of performance for concurrent transactions with different levels of security.

Keywords. Multilevel Security, Distributed Database, Concurrency, Transaction, Performance.

8.1 Introduction

The importance of databases in modern businesses, public and private organizations, banks, educational institutions and in general day-to-day applications is already huge and still growing. Many critical applications re-

quire databases. These databases contain data of different degree of importance and confidentiality, and are accessed by a wide variety of users. Integrity violations for a database can have serious impact on business processes and disclosure of confidential data. Traditional security provides techniques and strategies to handle such problems with respect to database servers in a non distributed environment. In a global enterprise the database access is required round the clock. The Database Management Systems (DBMS) has coincided with significant developments today in distributed computing and concurrent access technologies and this becomes the dominant data management tools for highly data intensive applications. A distributed database is a collection of multiple logically interrelated databases distributed over a computer network. A distributed DBMS is defined as the software system that permits the management of the distributed database and makes the distribution transparent to the users [1]

In this scenario, Data Base Management Systems (DBMS) are designed to meet the requirements of performance, availability, reliability, security and concurrency. The recent rapid proliferation of Web-based applications and information systems has further increased the risk exposure of databases in a distributed environment and hence the data protection is more crucial than ever. The data needs to be protected not only from external threats but also from insider threats [2]. The solution to data security is classified into three major categories: the protection of data against unauthorized disclosure, prevention of unauthorized and improper modification and prevention of denial of services. In addition to the different levels of security, the DBMS is to provide concurrent access of highly available data in the presence of large and diverse user populations. Therefore it is obvious that multilevel security must be provided to the DBMS mainly on distributed environment. Concurrency control is an integral part of the database systems. It is used to manage the concurrent execution of operations by different transactions on the same data with consistency. Several methods have proposed to provide secure concurrency control to achieve correctness and at reduced cost of high security level transactions. One of the most important issues for concurrency control in MLS database system is the cover channel problem [3]. It naturally comes due to the contention for the shared data items by transactions executing at different security levels. The most common instances of totally ordered security levels are the Top-Secret(TS), Secret(S), and Unclassified(U) security levels encountered in the military and government sectors. In this paper, it is proposed a model to enhance the performance of concurrency for Multilevel Secure Distributed Database System. This model allows users to access a database concurrently from geographically dispersed locations through use of concurrency control locking algorithm.

The paper is organized as follows: The review of literature on MLS distributed database and related issues on concurrency control algorithms are presented in the Section 8.2. Section 8.3 presents the proposed multilevel secure distributed database model. Section 8.4 presents lock based protocol for the proposed concurrency model. Section 8.5 concludes paper.

8.2 Review of literature

The operations of the database can be performed in the form of transactions. Several units of works that form a single logical unit of work can be called as a transaction. This can be performed under supervision of transaction manager. A valid transaction must be satisfied the Atomicity, consistency, Isolation, Durability properties [3]. The transaction manager can allow two or more transactions to access the same data called concurrent transactions. The un-controlled concurrent transaction leads to inconsistent database and violate the isolation property. Hence, the transaction manager needs to control the interactions between the transactions. To achieve this task, transaction manager uses several concurrency control mechanisms, such as locks, time stamps, etc.

James M. Slack [3] presented two security mechanisms: the first mechanism is based on one-way protected group. One-way protected group is a set of one-way protected objects. Each one-way protected object in the group will accept messages only from a distinguished object in the group called the interface object. A one-way protected group supports data integrity and access integrity. The second mechanism is two-way protected group. It is an extension of one-way protected group. Each object in the group could only send the messages of that group. This model fails to address how security and integrity policies are implemented with protected groups.

Keishi Tajimia [4] has developed a technique to statically detect the security flaws in OODBs. He designed a framework to describe the security requirements and developed an algorithm to determine the security flaws. In this technique a user can bypass the encapsulation and abuse the primitive operations inside the functions and also the properties of aggregate are not given.

Linda M. Null et al. [5] have defined security policy for object-oriented data model. This security policy addresses mandatory as well as discretionary access controls. This model is based on the classes derive security classification constraints from their instances and logical instances. The implementation and suitability of these policies in the object-oriented envi-

ronment is not presented in their proposal.

Sushil Jajodia et al. [6] have proposed a database security model for mandatory access control that details with the Object-Oriented Data model. This includes a message filtering algorithm that protects the illegal flow of information among objects of various security levels. Finally a set of principles is defined to design and implement security policies in Object-Oriented Database Management Systems. This model fails to address information flow are rendered explicitly.

Roshan. K. Thomas et al. [7] give a kernelized architecture for multilevel secure Object-Oriented Database Management Systems (DBMS's) which support write-up. However, supporting write-up operations in object-oriented systems is complicated by the fact that such operations are no longer primitive, but can be arbitrarily complex and therefore can take arbitrary amounts of processing time. This architecture supports Remote Procedure Call (RPC) based write-up operations. Dealing with the timing of such write-up operations consequently holds less time.

Bertino et al. [8] proposed the practical relevance of nested transactions and for the theoretical issues related to the development of suitable locking mechanisms and serializability theories for nested transaction. In particular, the interactions between parent and child transactions in the same nested transactions can be executed concurrently. This require revise and extend primitives and the locking protocol, however as it mentioned. Serializability theory for nested transactions is substantially more complex.

Lin et al. [9] a simplified simulation model is used to compare the performance of basic timestamp, multi version timestamp, and two-phase locking algorithms. It does not include different data distributions(partitioned, replicated, etc.), and simplified communication delay by combining CPU processing time, communication delays, and I/O processing time for each transaction.

Navdeep Karur et al. [10] have presented a simulation model of a multilevel secure distributed database system using secure concurrency algorithm. It addresses the performance price paid for maintaining security in a MLS/DDBMS, but performance of higher security level transactions in a replicated database has not been studied.

Having studied the above literature, it has been identified that most of the research efforts in the area of secure concurrency control are focused on centralized databases. Concurrent access of distributed database is mandatory for applications like banking, financial, enterprises, industry and institutes etc., Concurrent Transactions within a distributed database management system face several restrictions. The proposed model is designed to perform efficient, secured concurrent transactions based on subquery analyzer.

8.3 Computing complexity

The data stored in a database should be secured from the unauthorized users. The data retrieval time is minimized by the Lock Manager (LM). This manager handles the locking mechanism for distributed sites. All transactions are sorted through the security manager (SM). The security manager (SM) handles the authentication of the user using with proper verification. The different query levels proposed in the model are View level ($SL(D_u)$), Secret level ($SL(D_s)$) and Top secret level ($SL(D_t)$). If the user is permitted access to the view level, the query is limited with the view level transaction ($SL(Q) < SL(D_u)$). Otherwise, the query is roll back to Transaction Level security. Transaction level is also classified into secret and top secret respectively upon the user classifications.

The query optimizer further divides the query into various levels for distributed access and creates a new optimized query according to the data distribution. The Transaction Manager (TM) pools the query in the transaction queue and allows the transaction to be executed according to the load balance of the transaction concurrency. The network traffic is also considered by the Transaction Manager (TM). The Transaction Concurrency Manager (TCM) again analyzes the query and arranges the query for various levels of concurrency. The waiting time stamp is used for each transaction. The transactions are executed without concurrency when the waiting time stamp is expired. So the infinity waiting time of transaction is avoided. The waiting time stamp for each transaction is proportional to the security levels. The lock manager transmits the lock signal to the entire distributed database sites. It allows the transaction to update the data only when there is no objection from any other sites. Otherwise, it rolls back the transaction. If there is no objection from all other lock managers then it locks the data. The transaction is rolled back by the lock manager if the transaction exceeds the timestamp limit. Thus the proposed model facilitates to access data only after passing through multiple security levels and also allowed to access concurrently without any access conflict. The proposed model is depicted in Figure 8.1.

8.3.1 Secure Concurrency Control Protocol

The various modules of the secure concurrency control protocol including the concurrency components are illustrated as follows:

The various modules proposed in the MLS framework including concurrency components are briefly presented here. The User Module (UM) handles the data for the accepted users with their privileges. It is defined as a

three tuple. $U=(V_s, D, M)$, where V_s is a verification code set, D is the data which is associated with the user and M is the mode of accessibility. The set V_s may be any one of security mechanism such as Username and password, IP based security and other hybrid security. User Log (UL) is maintained by the User Module (UM).

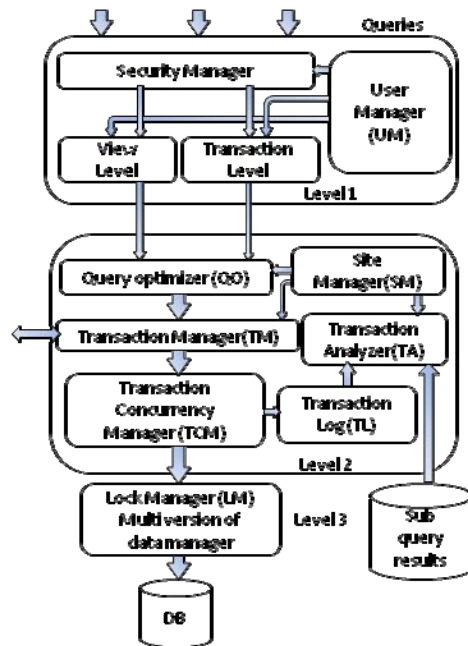


Fig. 8.1. Proposed Model

The data stored in a database should be secured from the unauthorized users. The security manager (SM) handles the authentication of the user using proper verification. All transactions are sorted through the security manager (SM). The different query levels proposed in the model are View level ($SL(D_u)$), Secret level ($SL(D_s)$) and Top secret level ($SL(D_t)$). If the user is permitted access to the view level, the query is limited with the view level transaction ($SL(Q) < SL(D_u)$). Otherwise, the query is rolled back by the Security Manager (SM). Transaction level is also classified into secret and top secret respectively based on the user classifications. In the transaction level one can update the data, but updation of data is not permitted in the view level. Each user having their own level of access limits to the data. Each transaction is permitted with the access limits controlled by the Security Manager (SM).

The Query Optimizer is designed to optimize the queries received from distributed locations. Let the set of sites (S_i) and each site S is having the set of fields (F_j). The fields F is the sub set of sites S . Let T_i be any transaction which involves the set of sites S_i and output or condition fields TF_i . The site S_i is eliminated from the query, if any one of the fields F_j of S_i does not belongs to the set TF_i . This process will eliminate the unnecessary sub transaction ST_i . So the network and data computation cost is reduced. The procedure is presented below.

```
void queryOptimizer(transaction T)
    TF[]=getOutputFields(T)
    TF[]=getComputationalFields(T)//add to output fields
    S[]=getDistributedDatabase(T)
    F[][]=getFields(S) //two dimensional table match the each row
    F[x][] with TF[]
    If F[x][] is not match with any TF[]
        eliminate S from T
```

The Site Manager (SMR) looks after all the distributed data which is created or configured for the transaction. This helps to secure the data from the unauthorized network access to the services. The site manager sends the signal to its own network boundary about the various locations of which data distributed and its configuration. This helps to attach and detach of sites over the network. The site manager has the details about the sites (S_i) and the associated fields (F_i). A transaction T_i is rolled back, if the transaction wants to access field TF_i in site S_i , which is not belongs to S_i . The Site Manager (SMR) handles the distributed sites and their locations. It also handles the frequency of updating the records. So it assigns a time stamp for each site. This time stamp is used for the transaction analyzer for flush the Sub Query Results (SQR).

The Transaction Manager (TM) further divides the query into various levels for distributed access and creates a new optimized query according to the data distribution. The Transaction Manager (TM) pools the transactions T_i in the transaction queue and allows the transaction to be executed according to the load balance of the transaction concurrency. The network traffic is also considered by the Transaction Manager (TM). Let T_i be any transaction and ST_{ij} be any sub transaction which is derived from the parent transaction T_i . We can define dispatcher as a four tuple $T_i = \{ST, TS, DB, CL\}$. Let ST be the sub transaction, TS be the time stamp, DB be the database and CL be the listing port for the sub transaction in the network. The transaction T_i is rolled back, if its all sub transactions S_i is not finished with in the timestamp order. The transaction manager (TM) saves the sub query result in the disk until it receives the updated message from the da-

tabase. So before sending the Sub Transaction (ST) it checks its own Sub Query Results (SQR). For that reason we can avoid repeated transaction for the same data. The Transaction Dispatcher (TD) sends the sub query to various sites. After completing that query it assembles the result and produces the result. The dispatcher using the timestamp mechanism for getting the results back.

The Transaction Analyzer (TA) analyzes the pooled transactions arrived from various distributed locations using the transaction log and creates the Frequently Accessed Fields (FAF) table. It also set the expiry time for each record in Frequently Accessed Fields (FAF) table. Using the timestamp, the FAF table entry is flushed. Let T_i be any transaction, TF_i be a set of fields which is going to be access by the transaction T_i , FAF_i be any entry in FAF table, $FAFT_i$ be the time stamp for record FAF_i , $FAFS_i$ be the current state of the Sub Query Results (SQR). The sub transaction ST_i for any transaction T_i is allowed to access the other site, if there is no entry in FAF table or the current state of $FAFS_i$ is cleared. The Sub Query Results (SQR) is updated, if or the current state of $FAFS_i$ is cleared and the sub transaction ST_i for any transaction T_i is allowed to access the other site. Transaction Analyzer (TA) flushes the Sub Query Results (SQR) when it receives the updated message from any other site. The Sub Query Results (SQR) is only for the transactions which are frequently involved. SQR proposed in this model reduce the unnecessary data access time and wait time for every transaction.

The Transaction Concurrency Manager (TCM) analyzes the concurrent transactions T_i and arranges the transaction for various levels of concurrency TC_i . The waiting time stamp is used for each transaction. The transactions are executed without concurrency when the waiting time stamp is expired. Thus the infinite waiting time of transaction is avoided. The waiting time stamp for each transaction is proportional to the security levels.

Let $T_1, T_2, \dots, T_n$ be n transactions which can be executed concurrently. Let U_i be the unit of a transaction for any transaction T_i . The schedule S is created for the transactions $T_1, T_2, \dots, T_n$. The schedule S can be defined as a two tuple. $S = (U, T)$, where U is a unit and T is a transaction. Let Unit Tree (UT) be the executed unit of transaction which are waiting for commit. The node for the unit tree is constructed when each unit transaction U is executed by the concurrency manager. The Transactions (T_i) are associated with the one-to-one mapping to the unit tree UT. The state for the Unit Tree (UT) is marked as roll back tree when the unit transaction of Schedule is failed. The Transaction associated with UT_i will not be executed in the Schedule if the Unit Tree UT_i is set by the Concurrency Manager (CM). The final commit is only after finishing the entire schedule. The commit is only for the Unit Tree which are not marked as a roll back

tree by the Concurrency Manager (CM). The According to this new concurrency algorithm, If the schedule S is failed then there is no need to roll back the entire transaction.

The Lock Manager (LM) transmits the lock signal to the entire distributed database sites. It allows the transaction to update the data only when it passes all the clearance from all security levels. Otherwise, it rolls back the transaction. If there is no objection from all other lock managers then it locks the data. The transaction is rolled back by the lock manager if the transaction exceeds the timestamp limit. The locking mechanism follows a **non cyclic** tree structure. If here any cyclic tree is formed by the transaction T_i then the T_i is rolled back. Hence the deadlock is prevented. Thus the proposed model facilitates to access data only after passing through multiple security levels and also allowed to access concurrently without any access conflict.

8.3.2 Concurrency control algorithm

Distributed database management system (DDBMS) allows users to access a database concurrently from geographically dispersed locations interconnected by a network. Concurrent accesses to the database have to be synchronized in order to maintain data consistency and to ensure correctness. This is achieved through use of distributed locking protocol which is applied in this proposed model. The efficient method of implementing concurrency is depicted in Figure 8.3 and presented in this section.

Let T_u denotes the unclassified security level transaction. Let T_s denotes the secret security level transaction and T_{ts} denotes the top secret security level transaction, i.e. $SL(T_u) \leq SL(T_s) \leq SL(T_{ts})$.

Let $r(x)$ and $w(x)$ be the read and write of data item x respectively.

Based upon the above assumption, the following conflicts may occur:

1. (Read-down conflict among different levels): Read-down conflict occurs between $SL(T_s) \leq SL(T_{ts})$'s read operation, $r[x]$, and $SL(T_{ts})$'s write operation, $w[x]$.
2. (Read-write conflict at same level): Read-write conflict occurs between $SL(T_s)_i$'s read operation, $r[x]$, and $SL(T_s)_j$'s write operation, $w[x]$. Where $SL(T_s)_i$, $SL(T_s)_j$ and x are at the same security level.
3. (Write-write conflict at same level): Write-write conflict occurs between $SL(T_{ts})_i$'s write operation, $w[x]$, and $SL(T_{ts})_j$'s write operation, $w[x]$. Where $SL(T_{ts})_i$, $SL(T_{ts})_j$ and x are at the same security level.

Every transaction in this security model must obtain a read lock before reading a data item and a write lock before writing a data item. The securi-

ty model allows a transaction to issue read-equal, read-down and write-equal operations. This is sufficient to prove that security is not violated through data access. The execution of a distributed transaction T is divided into sub-transactions T_i , where $i=1$ to n . A sub-transaction T_i is sent to the node N_i where the data is available and executed under the local security and concurrency transaction manager. If a sub-transaction fails, then the parent transaction is rolled-back and restarts after some delay to avoid repeated restart. The TLM (Transaction Lock Manager) determines at which node data items requested by a transaction are located. If the data is available in the parent node N_i , it is accessed in the same node N_i otherwise, if there is no local copy and multiple copies exist at more than one node, then one copy is randomly selected and locks other copies of the same data. It creates one sub-transaction for each node N_i that needs to be visited and acts as the coordinator in the distributed two-phase commit process. Even though the dispatches of sub-transactions of a transaction appear sequential, they are dispatches concurrently. Parent transactions originate from a fixed number of terminals and their number in the system is the sum of terminals connected to each node.

The CTM (Concurrency Transaction Manager) coordinates concurrency control activities with other nodes. In the case of data replication, it implements a read-one-write-all policy for read requests. For a write request, it consults all nodes that hold a copy of the desired data item. A DM (Data Manager) at every node contains information about data distribution and replication.

Let $V1$ be the set of transactions $V1=\{T1, T2, \dots, Tn\}$ and $V2$ be the set of Distributed Databases $V2=\{DB1, DB2, \dots, DBn\}$. The set of all links $E=\{e1, e2, \dots, em\}$ connecting from T_i where $T_i \in V1$ to DB_j where $DB_j \in V2$ denotes the transaction T_i can access the data on the distributed database DB_j as illustrated in the Figure 3.

A bipartite graph is a triple $G = (V1, V2, E)$ where $V1$ and $V2$ are two disjoint sets of vertices, respectively the top and bottom vertices, and $E \subseteq V1 \times V2$ is the set of edges.

The difference with classical graphs lies in the fact that edges exist only between top vertices and bottom vertices.

Two degree distributions can naturally be associated to such a graph, namely the **top degree distribution** ($V1_k$)

$$V1_k = \frac{|\{t \in V1 : d(t) = k\}|}{|V1|} \quad (8.1)$$

Where $d(t)$ denotes degree of a vertex t and the **bottom degree distribution** ($V2_k$),

$$V2_k = \frac{|\{t \in V2 : d(t) = k\}|}{|V2|} \quad (8.2)$$

A transaction cannot request additional locks once it has issued an unlock action. It holds on to all its locks (read or write) until it completes. A top secret security level transaction must release its read lock on a low data item when a unclassified security level transaction requests a write lock on the same data item and the aborted top secret security level transaction is restarted after some delay. Thus multiple transactions are performed simultaneously with minimum cost.

Table 8.1 shows the access permissions in the proposed model.

Table 8.1. Access permissions in the proposed model

	Data Items	SL(x_{ul})	SL(y_{sl})	SL(z_{ts})
Transactions				
SL(T_{ts})		r[x]	r[y]	r[z],w[z]
SL(T_{sl})		r[x]	r[y],w[y]	-
SL(T_{ul})		r[x]	-	-

Concurrency Control Algorithm

```

void concurrency(transaction T)
U[] = divide the transaction into various unit of transaction
Analyze the transactions, which are in the queue
Create the Schedule for the various units of transaction
While(schedule finish)
    U = next unit to be executed
    UT = the Unit Tree associated with U
    If (UT is not marked as roll back tree)
        Execute the unit (U) in the schedule
    If(U is success)
        Place U in the Unit Tree (UT)
    else
        mark the Unit Tree (UT) as roll back tree
    while (all unit tree UT in CM is traversed)
        if (UT is not marked as roll back tree)
            commit the Unit Tree UT
    else
        reject the transaction Ti which is associated
        with Unit Tree UT
    
```

8.4 Simulation results

The protocols for evaluating the performance of concurrency control are tabulated. This evaluation is based on the performance presented in [11]. The aim of this experiment is to test the transaction performance with the proposed concurrency algorithm. The model is simulated in a real time environment presented in Table 8.2.

Table 8.2. Simulation parameters

Parameter	Value
NumDBS	5
No. Query / sec	31
No. CPU	5
Disk for each site	5
Log disk	1
Concurrent Transaction / sec	11
Write ratio	6
Read ratio	25
Waiting time out transaction	4
Execution time	0.0715 s
Schedule Execution time	0.2015 s
Transaction size	10 records
CPU time / unit	0.0119 s
Network delay	0.01 s
Time for optimize	0.002
Time to partition	0.014
Unit / transaction	6

The performance study is carried out with varying time factor. The number of transactions is directly proportional to the increased level of security with varying time factor. Let NT be the number of transactions and TT be the time to finish the transactions then,

$$NT \propto 3.525 TT \quad (8.3)$$

The time for executing the transaction is directly proportional to the security level also. Let SL be the security level and TT be the time to finish the transaction then,

$$TT \propto 1.1928 SL \quad (8.4)$$

The simulation results graphically represented in Figures 8.4-8.6 show that the performance of concurrent transactions increased with the transactions arrival rate is increased. The performance of the transaction is high when the number of distributed database is increased.

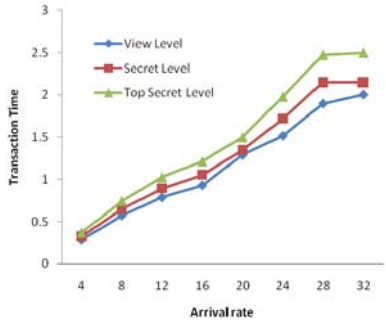


Fig. 8.4. Transaction time vs Arival rate

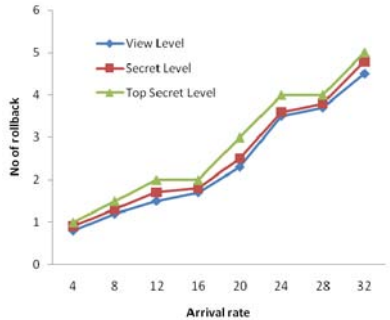


Fig. 8.5. No of rollback vs Arival rate

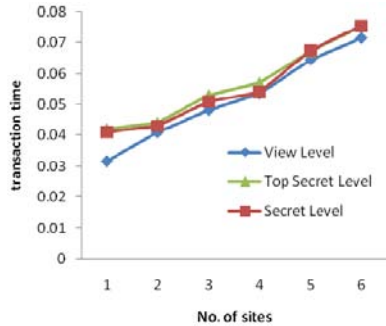


Fig. 8.6. Transaction time vs No. of sites

8.5 Conclusion

Distributed database systems today play a reality. Several organizations are now deploying distributed database systems. Security is a serious con-

cern a while accessing data. Different security levels need to be integrated into the database to avoid access conflict. We have proposed a new model for Multilevel Secure Real-time Database Systems (MLSDD) and deployed concurrency control for the secured access of data. The simulation study reveals that the performance of concurrent transactions is enhanced with multiple levels of security over the distributed database. This model can be applied for any real-time environment such as, corporate, financial enterprises, academic institute etc., performing day-to-day transactions in a distributed environment cutting edge to different levels of security.

References

1. Bertino Elisa, Sandhu Ravi (2005) Database security – concepts, approaches, and challenges. *IEEE Transactions on Dependable and Secure Computing* 2(1)
2. Tamper Ozsu M, Valduriez Patrick (1996) Distributed and parallel database systems. *ACM Computing Surveys* 28(1)
3. Slack James M (1993) Security in an object-oriented databases. *Proc. of ACM Transactions*, pp 155-159
4. Tajimia Keishi (1996) Static detection of security flaws in OODBMS. RIFMS, Kyoto University Japan. In: *ACM SIGMOD International Conference on Management of data SIGMOD '96*, 25(2):341-352
5. Null Linda M (1992) The DIAMOND security policy for object-oriented databases. In: *ACM annual Conference on Communications*, pp 49-56
6. Jajodia Sushil, Kogan Boris, Sandhu Ravi S A Multilevel Secure Object-Oriented Data Model. In: *Proceedings of ACM SIGMOD*, pp 596-616.
7. Thomas Roshan K, Sandhu Ravi S (1993) A kernelized architecture for multi-level secure object-oriented databases supporting write-up. *Journal of Computer Security*
8. Bertino E, Catania B, Ferrari E (2001) A nested transaction model for multi-level secure database management systems. *ACM Transactions on Information and System Security* 4(4):321–370
9. Lin W, Nolte J (1983) Basic timestamp, multiple version timestamp, and two-phase locking. In: *Proceedings of 9th International Conference on VLDB Conference*, Florence, Italy
10. Kaur Navdeep, Sarje AK, Misra Manoj (2004) Performance Evaluation of Concurrency Control Algorithm for Multillevel Secure Distributed Databases. *IEEE Compter Society*
11. Xeoing Ming et al. (2002) Mirror: A state consciious concurrency control protocol for replicated real time database. *Journal of Information Systems* 27:277-297

Chapter 9

A new framework for non-deterministic multivalued system minimization

Adrian Zafiu, Ion Stefanescu

Department of Electronics, Communications and Computers, University of Pitesti, 110040 Pitesti Str. Targul din Vale, nr.1, Arges, Romania
<http://www.upit.ro>, adrian_zafiu@yahoo.com, adrian.zafiu@upit.ro
jean@upit.ro, ionjean@yahoo.com

Abstract. The main goal of this paper is to presents a new minimization method for the non-deterministic networks. The motivation of this article is given by the recently developed minimization system named COMIN. This method comprises a deep analysis of the deterministic and nondeterministic multivalued networks, complete or incomplete specified networks and of the possible adopted strategies during the combinational minimization.

Keywords. Multi-valued combinational minimization, generative method, processing redundancy, evaluation; implicants, network and parallel minimization, deterministic and nondeterministic networks

9.1 Introduction

Multivalued logic, as a tool of expressing and optimizing the input-output digital correspondences, expressing the decisional parts of digital systems and algorithms, becomes of a major importance in the field of software minimization, as well in the field of new developments of digital circuits, like current and optical implementation of gates and interconnection means. The field of interest is logical design and optimization of hardware

and software decisional systems. A natural way of expressing a function of a decisional system is a multivalued or a binary value table. Thus, the optimization kernel of a decisional system, either an algorithm logical structure, or a digital circuit structure, becomes the minimization of the input-output correspondence, represented by a value table. Presented method use a different principle named discrimination [3]. According to this principle we search the set of difference between groups of input combinations with the same output.

9.2 Minimization tools

Here is a decisional part of a given algorithm:

```
int function(int x1,int x2,int x3){
    if ( x1==1 && x2==3 && x3==1 ) return 0;
    if ( x1==2 && x2==1 && x3==4 ) return 3;
    if ( x1==3) if( x2==5 && x3==7 ) return 9;
                  else if( x2==7 && x3==2 ) return 7;
    if ( x1==4 && x2==2 && x3==2 ) return 0;
    if ( x1==5) if( x2==7) if( x3==2 ) return 7;
                  else if ( x3==3 ) return 0;}
```

The above algorithm could be written as a multivalued input-output correspondence table and the result of reconverting the optimized table into an algorithm is:

```
int function(int x1,int x2,int x3){
    if( x1==3 ) return 9;
    if( x1==4 ) return 3;
    if( x2==7 && x3==2 ) return 7;
    return 0;}
```

There are known minimization tools, like ESSPRESO, SIS, MVSIS, MVSIS-BLIF, and BALM (the later one, for some steps of automata minimization [6]). As far as we know at this moment, the main reference in the field of multivalued decision table minimization is the program MVSIS 3.0 [1, 2], but in many cases the results not represents an optimum, especially in the field of network minimization, as well as in the case of pure multivalued input-output deterministic and non-deterministic correspondences, and in the case of correlated outputs.

The principles and programs recently developed at the University of Pi-testi [3-5, 7-9] solve the most of the mentioned problems, offering a multivalued minimization technique, based on a different principle of minimiza-

tion - the discrimination principle [3, 4, 7] - and on an original generative minimization method [8], as well on an original network strategy [5].

The non-deterministic systems are minimized in two ways:

- full (n/n) non-determinism, by considering all initial options (i.e. preserving in the minimized solution all values of an output variable corresponding to a given input combination in the non-minimized table – which represents a non-deterministic correspondence, having a deterministic semantics),
- partial (k/n) non-determinism, by selecting an optimum (from the minimization point of view) subset out of the set of values of an output variable corresponding to a given input specification (having real non-deterministic semantics).

There is an option of minimizing system containing correlated outputs as well a selection of a default output value for deterministic and non-deterministic tables.

9.3 The proposed principles of minimization

The basic principles of the minimization techniques used in our works are mentioned:

Discrimination principle [3, 7]: according to this principle, there are preserved - in the input part of each row of a value table - the minimum number of specified positions, so that the row is still different (discriminated) from all other tables rows having different output values.

Preserving the specification integrity: the result of a minimization (the minimized table) preserves all the specifications of the initial input-output correspondence (of the initial table). Thus, when the specifications of an initial row are applied to the simplified table, then there are obtained the same output values as in the initial deterministic or nondeterministic non-minimized table;

Minimization on output value groups: the results of a multivalued decision system produce decisions for delivering each output value. Thus, the minimized outputs are, in fact, the value outputs.

Deterministic meaning of the non-univocities: each non-deterministic table having implicit non-univocities is converted into a table having explicit non-univocities only, and then expanded on the values of its output variables.

Deterministic semantics of non-univocities: An explicit non-univocity is considered as a set of simultaneous values corresponding to a given input specification row, if not otherwise specified (so called “non-

determinism with deterministic semantics”, or “full non-determinism”, denoted by n/n) - the reason is to preserve all initial intentions of the table designer.

Parallel or network minimization of multiple outputs tables: tables with multiple outputs could be minimized in two ways:

- parallel: each output is minimized separately, independent of the other outputs;
- network: the output variables can be connected as inputs to other outputs, resulting simpler minimized results, due to supplementary information offered by these interconnections;
- network minimization oriented on the values of output variables:

Preserving the correlations among different outputs, or groups of outputs in a non-deterministic network table: in case of a non-deterministic table having declared correlated outputs, the minimization must consider the aspect and produce a minimized table preserving all correlations implemented into the non-minimized table.

A single version of minimized solution: the minimization result presents one version of solutions out of a class of equivalent minimized solutions.

9.4 Network minimization of non-deterministic tables with correlated outputs

Network strategy includes an option selecting a minimization preserving all value correlations between different output variables, as they are manifested in the initial non-minimized table. The option has a meaning in the case of non-deterministic tables having more rows with the same input specification. In the rest, the correlations between outputs are automatically preserved during the minimization, regardless of the network or of the parallel type of minimization. If the correlation option is not activated, the network minimization could produce simpler results, because there are fewer constraints, but the correlations among the outputs are not, generally, preserved. For example, a part of a non-deterministic table, having two rows with the same input specification and two outputs with different non-univoque specifications, is given in next table:

Table 9.1.

x1	x2	q1	q2
2	0	1	(1,2)
2	0	(1,2,3)	3

As a result of minimization, there are two solutions, depending on the initial intentions of the designer:

- outputs q1 and q2 are independent, not informationally correlated, in which case the above table could be condensed, as in Table 9.1, implying nine possible (q1 q2) output pairs corresponding to input (2 0), namely: (1 1), (1 2), (1 3), (2 1), (2 2), (2 3), (3 1), (3,2), (3 3) (see Figure 9.1)

Table 9.2.

x1	x2	q1	q2
2	0	(1,2,3)	(1,2,3)

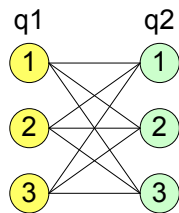


Fig. 9.1. Output interconnections in a non-correlated minimization

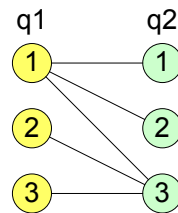


Fig. 9.2. Selective output interconnections in a correlated minimization

- outputs q1 and q2 are informationally correlated, which means the only allowed output combinations are those directly resulting out of Table 9.1, and no more, namely: (1 1), (1 2), (1 3), (2 3), (3 3).

Results with uncorrelated and correlated outputs, full non-determinism (n/n):

Table 9.3.

x	y	z	/	a	y	/	B
0	-	-	/	1	1	/	{3,4}
0	-	1	/	{1,2}		/	2
1	0	-	/	2			
			/	0			

Table 9.4.

x	Y	b	/	a	y	/	b
0	-	-	/	1	1	/	{3,4}
0	-	3	/	{1,2}		/	2
1	0	-	/	2			
			/	0			

In the case of correlated outputs, variable „a” contains, as feedback, output „b”, thus satisfying the correlation between „a” and „b”. There is also noticed the uncorrelated outputs results are generally simpler than correlated outputs results.

9.5 Conclusions

Based on a different minimization principle – the discrimination [3, 7] – and on an appropriate original multivalued minimization algorithm [8], there is presented here an efficient technique to obtain powerful minimizations of binary and multivalued decisional systems, by using a strategy of network expansion and processing of the output nodes.

The discrimination does not need to consider explicitly the unspecified input combinations (“don’t care”).

The network structuring on output values is a natural one in the multivalued case, where the minimization and the results must be expressed on values of each multivalued output. The network strategy is recommended even in single multivalued outputs, by expanding the output variables on their values.

The non-deterministic input-output correspondences could be treated either integrally, by considering all non-deterministic specifications (full non-determinism), or by selecting a part of the non-deterministic specifications (partial non-determinism), in order to get a simpler solution.

Due to cross correlations appearing among different value outputs in a network minimization, all the interdependencies between outputs, settled by an initial statement of a nondeterministic decisional system (by the initial table), are automatically considered (if so specified), by means of a special option.

There is also implemented an option to select, if possible, a default value, taking account of non-deterministic constraints.

As presented in [8], the minimization method and algorithms do not depend exclusively on the number of input variables; the inner structure of the table decides the upper limits.

References

1. Gao M, Jiang J-H, Jiang Y, Li Y, Sinha S, Bryton R (2001) MVSIS. In: the Note of the International Workshop on Logic Synthesis, Tahoe City

2. Chai Donald, Jiang Jie-Hong, Jiang Yunjian, Li Yinghua, Mischenko Alan, Brayton Robert (2003) MVSIS 2.0 User's manual. University of Berkeley CA 94720
3. Ștefănescu Ion (2005) Discrimination: A new principle in the efficient minimization of the binary and multiple-valued functions. In: 1st Int Conf on Electronics, Computers and Artificial Intelligence, ECAI 2005, University of Pitești, Romania
4. Zafiu Adrian, Ștefănescu Ion (2005) Minimization aspects of hardware and software systems – an extension of the discrimination method. In: 1st Int Conf on Electronics, Computers and Artificial Intelligence, ECAI 2005, University of Pitești
5. Ștefănescu Ion, Zafiu Adrian (2007) An efficient network strategy in deep minimizations of deterministic and nondeterministic multivalued decisional systems. In: 2nd Int Conf on Electronics, Computers and Artificial Intelligence ECAI 2007, University of Pitești, Romania
6. Brayton Robert K¹, Jiang Jie-Hong R¹, Mishchenko Alan¹, Villa Tiziano², Yevtushenko Nina³ BALM User's manual. ¹University of California, Berkeley, ²University of Udine, Italy, ³Tomsk State University, Russia
7. Ștefănescu Ion (2006) Proiectarea logica a sistemelor decizionale hardware și software – Aspecte de bază. (Logic Design of Hardware and Software Decisional Systems – Basic Aspects – (in romanian)), Matrix Rom, București, p 630
8. Zafiu Adrian (2007) An efficient generative algorithm developed for the minimization of the multi-valued specifications. In: 2nd International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2007, University of Pitești, Romania
9. Zafiu Adrian, Ștefănescu Ion, Franti Eduard (2007) A new method for combinational minimization of multi-valued decisional system specifications. Services and Software Architectures, Bucharest, Romanian Academy, pp 27-37

Part 12

Web-based applications

Web based applications provide users with the opportunity to save time and money, and improve the way to interact with clients, suppliers and business partners. In this part are presented many Web application frameworks which facilitate rapid application development by allowing the programmer to define a high-level description of the program. In addition, there is potential for the development of applications on Internet Operating Systems, although currently there are not many viable platforms that fit this model. The papers cover different aspects: thread pool-based improvement of the mean-value analysis algorithm, security enhancements for web based applications, description and discovery of web applications in grid, user's preferences about personal data, peak analysis for efficient string sorting, parametric studies for the AEC domain using InteliGrid platform and an auction-based resource allocation strategy with a proportional share model for economic-based grid systems.

Chapter 1

Thread pool-based improvement of the mean-value analysis algorithm

Ágnes Bogárdi-Mészöly,¹ Takeshi Hashimoto,² Tihamér Levendovszky,¹
Hassan Charaf¹

¹Department of Automation and Applied Informatics, Budapest University
of Technology and Economics

Goldmann György tér 3. IV. em., 1111 Budapest, Hungary

²Department of Electrical & Electronics Engineering, Shizuoka University
5-1, 3-chome Johoku, 432-8561 Hamamatsu, Japan

Abstract. The performance of information systems is an important consideration. With the help of proper performance models and evaluation algorithms, performance metrics can be predicted accurately. The goal of our work is to improve the Mean-Value Analysis (MVA) evaluation algorithm based on the investigation of thread pools. In our work, the performance metrics of multi-tier information systems are predicted with the help of a queueing model and the improved MVA. The proposed evaluation algorithm has been implemented, the inputs have been estimated based on one measurement, and the model has been evaluated to predict performance metrics. In addition, ASP.NET web applications have been tested with concurrent clients in order to validate the proposed algorithm in different versions of ASP.NET environments.

Keywords. Queueing model, Mean-Value Analysis, Thread pool, Performance prediction, Web application

1.1 Introduction

One of the most important factors of information systems is performance, because they face a large number of users, and they must provide high-availability services with low response time, while they guarantee a certain throughput level. With the help of a properly designed performance model and an appropriate evaluation algorithm, the performance metrics of a system can be determined at early stages of the development process.

Several methods have been proposed in the past few years to address this goal. A group of them is based on queueing networks or extended versions of queueing networks [7, 11, 15]. Another group uses Petri-nets or generalized stochastic Petri-nets [1, 8]. The third group of the approaches uses a stochastic extension of process algebras, like TIPP (Time Processes and Performability Evaluation) [6], EMPA (Extended Markovian Process Algebra) [2], and PEPA (Performance Evaluation Process Algebra) [5].

Performance metrics depend on many factors. Several papers have investigated various configurable parameters, how they affect the performance of web-based information systems. Statistical methods and hypothesis tests are used to retrieve factors influencing the performance. An approach [14] applies analysis of variance, another [3, 4] performs independence test.

Today one of the most prominent technologies of information systems is Microsoft .NET. The application server has several settings which can affect the performance [10]. The request of the client goes through several subsystems before it is served as depicted in Figure 1.1.

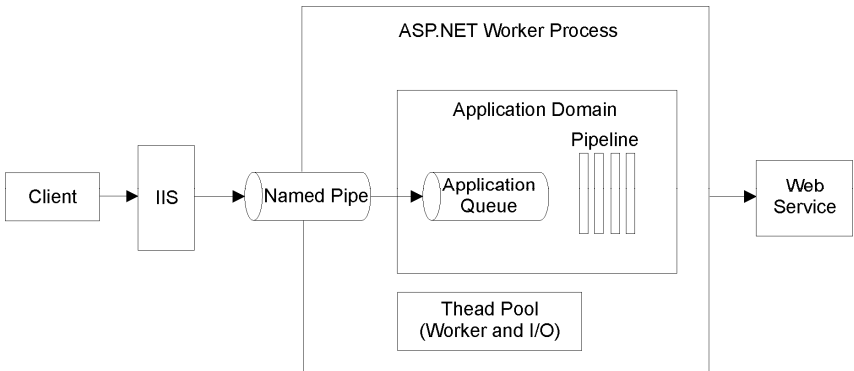


Fig. 1.1. The architecture of ASP.NET environment

From the IIS (Internet Information Services), the accepted HTTP connections are placed into a named pipe. This is a global queue between IIS

and ASP.NET, where requests are posted from native code to the managed thread pool. From the named pipe, the requests are placed into an application queue, also known as a virtual directory queue. There is one queue for each virtual directory. The global queue is managed by the process that runs ASP.NET, its limit is set by the *requestQueueLimit* property. The application queue limit is configured by the *appRequestQueueLimit* property. When the limit is reached, the requests are rejected. When an application pool receives requests faster than it can handle them, the unprocessed requests might consume all of the memory, slowing the server and preventing other application pools from processing requests.

The *maxWorkerThreads* attribute means the maximum number of worker threads, the *maxIOThreads* parameter is the maximum number of I/O threads in the .NET thread pool. These attributes are automatically multiplied by the number of available CPUs. The *minFreeThreads* attribute limits the number of concurrent requests, because all incoming requests will be queued if the number of available threads in the thread pool falls below the value for this setting. The *minLocalRequestFreeThreads* parameter is similar to *minFreeThreads*, but it is related to requests from localhost (for example a local web service call). These two attributes can be used to prevent deadlocks by ensuring that a thread is available to handle callbacks from pending asynchronous requests.

In our previous work [3, 4], the effects of the limits of the thread and the queue size have been statistically analyzed, and six influencing parameters of the performance have been found: *maxWorkerThreads*, *maxIOThreads*, *minFreeThreads*, *minLocalRequestFreeThreads*, *requestQueueLimit*, *appRequestQueueLimit*, which is proven by a statistical method, namely, the chi square test of independence.

The paper is organized as follows. Sect. 1.2 covers backgrounds and related work. Sect. 1.3 describes our improved MVA algorithm. Sect. 1.4 presents the experimental configurations, the performance prediction with the proposed algorithm and its validation in different versions of ASP.NET environments. Finally, Sect. 1.5 reports conclusions and future work.

1.2 Backgrounds and related work

Queueing theory [7] is one of the key analytical modeling techniques used for information system performance analysis. Queueing networks and their extensions (such as queueing Petri nets [9]) are proposed to model web-based information systems [11, 12, 15].

In [15], a queueing model is presented for multi-tier information systems, which are modeled as a network of M queues: $Q_1, \dots, Q_M$ illustrated in Figure 1.2. Each queue represents an application tier. A request can take multiple visits to each queue during its overall execution, thus, there are transitions from each queue to its successor and its predecessor as well. Namely, a request from queue Q_m either returns to Q_{m-1} with a certain probability p_m , or proceeds to Q_{m+1} with the probability $1-p_m$. There are only two exceptions: the last queue Q_M , where all the requests return to the previous queue ($p_M = 1$) and the first queue Q_1 , where the transition to the preceding queue denotes the completion of a request. S_m denotes the service time of a request at Q_m ($1 \leq m \leq M$).

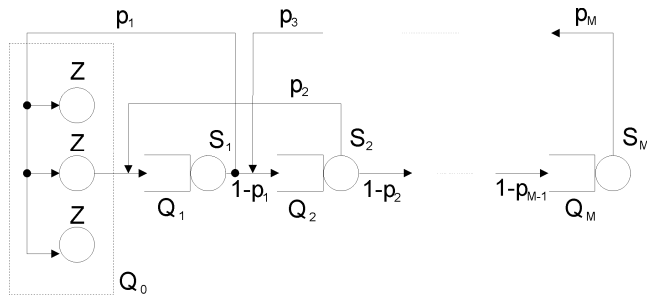


Fig. 1.2. Modeling a multi-tier information system using a queueing network

Internet workloads are usually session-based. The model can handle session-based workloads as an infinite server queueing system Q_0 that feeds the network of queues and forms the closed queueing network depicted in Figure 1.2. Each active session is in accordance with occupying one server in Q_0 . The time spent at Q_0 corresponds to the user think time Z .

The model can be evaluated for a given number of concurrent sessions N . A session in the model corresponds to a customer in the evaluation algorithm. The MVA algorithm for closed queueing networks [7, 13] iteratively computes the average response time and the throughput performance metrics. The input parameters of the algorithm are N the number of customers, M the number of tiers, Z the average user think time, V_m the visit number and S_m the average service time for Q_m ($1 \leq m \leq M$). The output parameters are τ the throughput, R the response time, R_m the response time for Q_m and L_m the average length of Q_m . The initialization is $L_m = 0$ ($1 \leq m \leq M$).

The algorithm introduces the customers into the queueing network one by one ($1 \leq n \leq N$). The steps of an iteration are the follows:

$$R_m = V_m \cdot S_m \cdot (1 + L_m), \quad (1.1)$$

$$\tau = \frac{n}{Z + \sum_{m=1}^M R_m}, \quad (1.2)$$

$$L_m = \tau \cdot R_m. \quad (1.3)$$

The cycle terminates when all the customers have been entered.

1.3 Improving MVA based on thread pool investigation

The MVA algorithm can be effectively improved to model the behavior of the thread pool. Consider that the actual request contains CPU as well as I/O calls. In case of multiple threads, I/O calls do not block the processing, because the execution can continue on other non-blocked threads. This enables handling I/O requests and executing CPU instructions simultaneously. Therefore, in the MVA algorithm, updating the queue length (shown in Eq. 1.3) must be modified in the input, output and input/output tiers, in the CPU tiers, this equation preserves its original form.

Proposition 1: In the input, output and input/output tiers Eq. 1.3 is the following:

$$L_{I/O} = \tau \cdot R_{I/O} - \frac{S_{CPU}}{S_{I/O}} \cdot L_{CPU}, \quad (1.4)$$

where the CPU index means a CPU tier index and the I/O index corresponds to an I/O tier index from $I \leq m \leq M$. Additionally, L_{CPU} is from the previous iteration step, namely, L_{CPU} is updated only after this step. After Eq. 1.4, one more checking step is necessary:

$$\text{if } L_{I/O} < 0 \text{ then } L_{I/O} = 0. \quad (1.5)$$

Proof: Consider that the CPU queue and the I/O queue contain some requests respectively. If a new request arrives, it must wait until each previous request has been handled in the CPU queue. It takes $L_{CPU} \cdot S_{CPU}$ time unit to handle each CPU request. During this time the requests residing in the I/O queue can be handled simultaneously, because the CPU and the I/O devices can work in parallel. If an I/O call occurs, the CPU has to wait until the I/O device handles the request. In case of multiple threads the CPU does not have to wait, because the execution can continue on another thread as depicted in Figure 1.3.

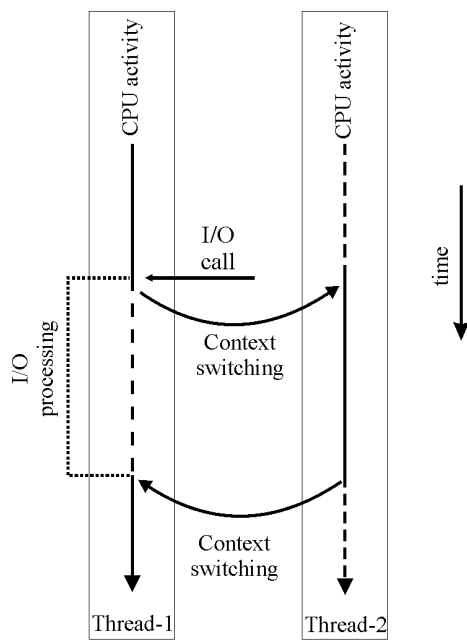


Fig. 1.3. Illustrating the behavior of multiple threads

Namely, during this time interval $S_{CPU} / S_{I/O} \cdot L_{CPU}$ I/O requests can be handled, which affects that the number of requests waiting in the I/O queue decreases. This difference should be subtracted from the number of requests waiting in the I/O queue calculated by the MVA.

After this equation one more checking step is necessary, because the average length of a queue cannot be negative. Thus, if the obtained $L_{I/O}$ value is less than 0, the $L_{I/O}$ must be equal to 0. $\square$

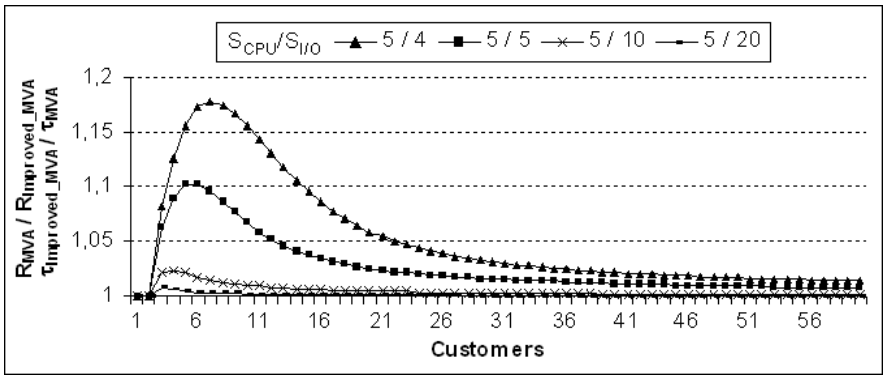


Fig. 1.4. The effect of the proposed coefficient

This extension has a considerable effect on the predicted response time and throughput if the S_{CPU}/S_{IO} coefficient is near to 1 or greater than 1 as illustrated in Figure 1.4. If this coefficient much smaller than 1, the outputs of the MVA and the proposed improved MVA are closer.

1.4 Applying improved MVA algorithm in different ASP.NET environments

Firstly, our experimental configurations are described. Secondly, our performance prediction with the proposed improved MVA algorithm is presented. Finally, our experimental validation of the proposed algorithm is described in different ASP.NET environments.

1.4.1 Experimental configurations

Three-tier ASP.NET test web applications have been implemented in different versions of ASP.NET environments (Figure 1.5). Compared to a typical web application, they have been slightly modified to suit the needs of the measurement process. The web applications were designed in a way that the input values of the model parameters can be determined from the results of one measurement. Each page and class belonging to the presentation, business logic or database was measured separately.

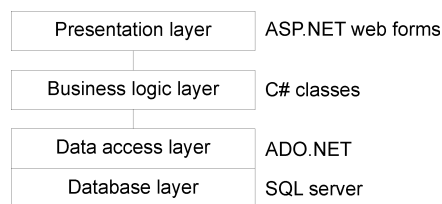


Fig. 1.5. The test web application architecture

The web server of our test web application was Internet Information Services (IIS) 6.0. The server runs on a 2.8 GHz Intel Pentium 4 processor with Hyper-Threading technology enabled. It had 1GB of system memory; the operating system was Windows Server 2003 with Service Pack 1. There were two environments. In the first environment, the application server was ASP.NET 1.1 runtime environment, the database management system was Microsoft SQL Server 2000 with Service Pack 3. In the second environment, ASP.NET 2.0 and Microsoft SQL Server 2005 with Service Pack 1 were used.

During our measurements the limits of the thread type (*maxWorkerThreads*, *maxIOThreads*, *minFreeThreads*, *minLocalRequestFreeThreads*,) and the queue size (*requestQueueLimit*, *appRequestQueueLimit*) were hold on their default settings. Hyper-Threading technology is enabled on the processor of the server. Thus, it seems to have two processors, although physically there is only one processor. Hence the first two parameter (*maxWorkerThreads* and *maxIOThreads*), automatically multiplied by two.

The clients ran on another PC on a Windows XP Professional computer with Service Pack 2. The supporting hardware was a 3 GHz Intel Pentium 4 processor with Hyper-Threading technology enabled, and it also had 1GB system memory. The connection between the computers was provided by a 100 Mb/s network. The emulation of the browsing clients and the measuring the response time were performed by JMeter, which is an open source load tester. Tests can be created on a graphical interface. Virtual users send a list of HTTP requests to the web server concurrently. Each virtual user inserts an exponentially distributed think time between its requests with a mean of 4 seconds. With the help of JMeter, the measurement process can be easily automated.

1.4.2 Performance prediction with the improved MVA algorithm

It is worth decomposing a web application to multiple tiers, and modeling it according to Figure 1.2, because the service time of each tier can be very different. The presentation tier (UI) is an output tier and the database tier (DB) corresponds to an input/output tier. Thus, the proposed improvement must be applied in these tiers. The business logic layer (BLL) corresponds to a CPU tier.

Firstly, the MVA and the proposed improved MVA algorithm for closed queueing networks have been implemented with the help of MATLAB. Secondly, the input values of the algorithms have been estimated in both environments from one-one measurement. The inputs of the script are the number of tiers M , the maximum number of customers (simultaneous browser connections), the average user think time Z , the average service times S_m and the visit numbers V_m for Q_m ($1 \leq m \leq M$).

During the measurement the number of tiers was constant (three). The maximum number of customers means that the load was characterized as follows: we started from one simultaneous browser connection, then we continued with two, until the maximum number (arbitrary, in this case it is set to 60) had been reached. Since JMeter can measure only the response

time of the three tiers without user think time between the subrequests, in the algorithms the average user think time between the requests is 0.

Finally, the model has been evaluated. The script computes the response time and the throughput up to a maximum number of customers in both environments (depicted in Figures 1.6 and 1.7).

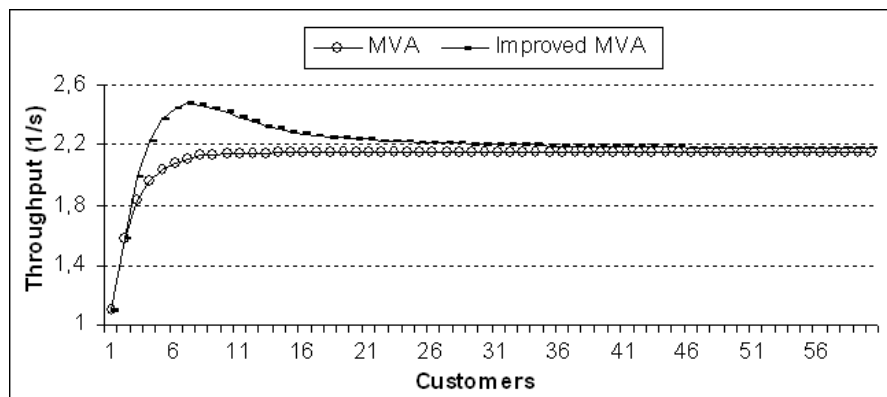


Fig. 1.6. The predicted throughput with MVA and with extended MVA in the first environment

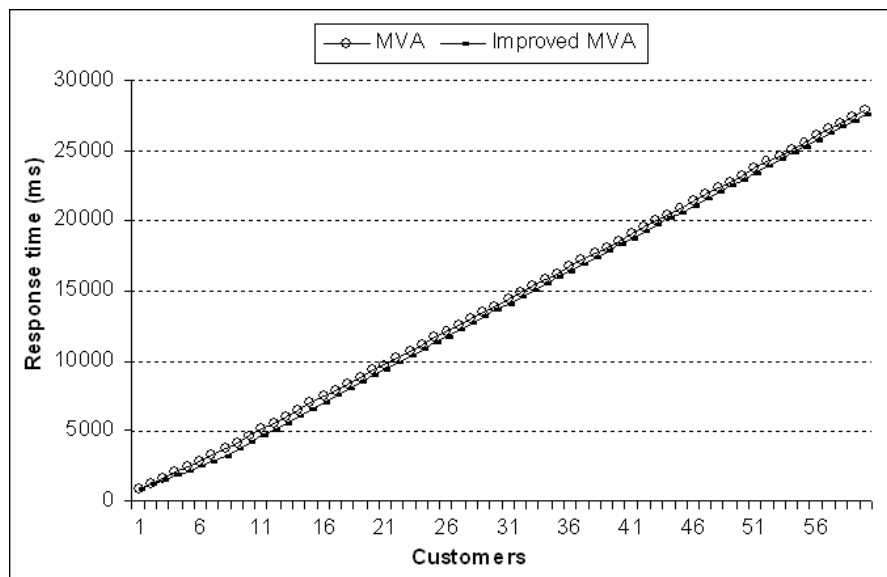


Fig. 1.7. The predicted response time with MVA and with extended MVA in the first environment

In these cases the proposed improvement has a considerable effect on the predicted response time and throughput in both environments, because the S_{BLL}/S_{UI} coefficient is near to 1. Thus, high curvatures can be observed in the response time and in the throughput as well.

1.4.3 Experimental validation

The proposed extended MVA algorithm has been validated in the two ASP.NET environments. Typical web applications (Figure 1.5) have been tested with concurrent user sessions, comparing the observed and predicted values in order to validate the proposed algorithm in both environments.

While the number of simultaneous browser connections varied, the average response time and throughput were measured in the two environments. As illustrated in Figure 1.8, the new version of ASP.NET runtime environment and SQL Server database management system are slightly faster than the older version, but the character of the curve shapes are very similar. The results correspond to the common shape of the response time and throughput performance metrics. Increasing the number of concurrent clients, the response time grows, since the queue limits are large enough, thus, all of the requests can be successfully served.

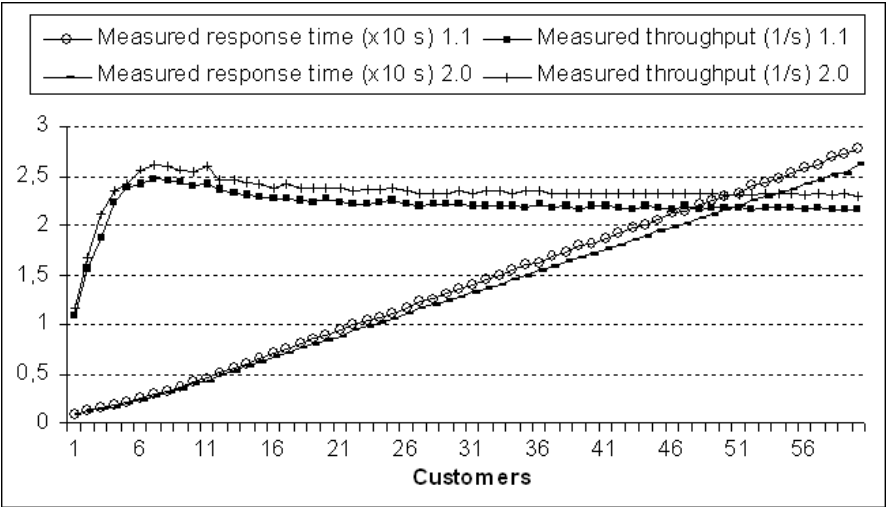


Fig. 1.8. The observed response time and throughput in ASP.NET 1.1 and 2.0 environments

Our proposed algorithm was experimentally validated by comparing the observed, the predicted values with MVA, and the predicted values with

our improved MVA in the two environments. Our results (depicted in Figures 1.9-1.12) have shown that the output of the proposed algorithm approximates the measured values much better than the MVA considering the shape of the curve and the values as well in both environments. Thus, the proposed algorithm predicts the response time and throughput much more accurate than the MVA algorithm in both environments.

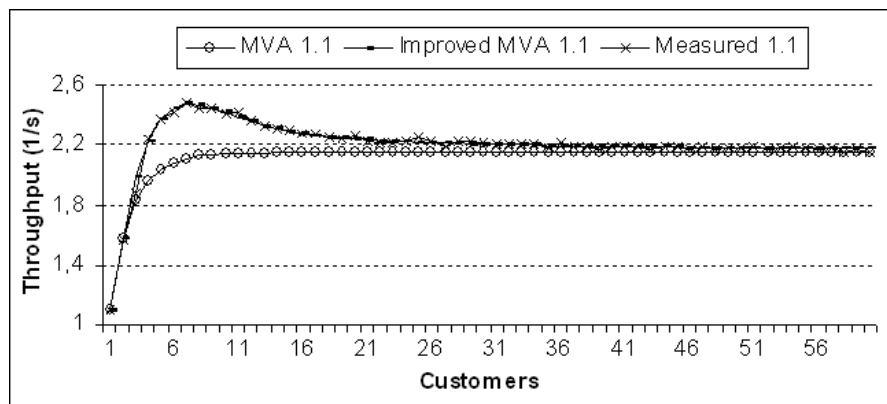


Fig. 1.9. The observed, the predicted with MVA and the predicted with improved MVA throughput in the first environment

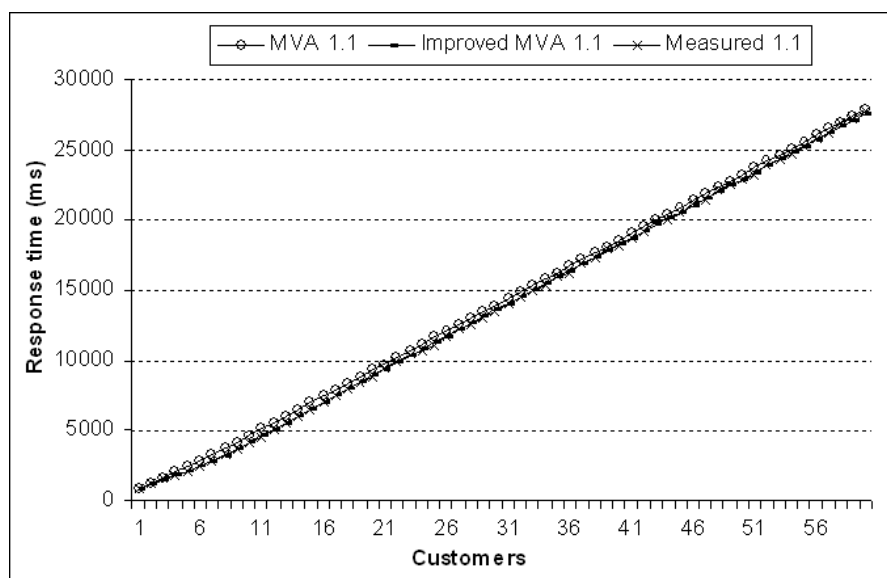


Fig. 1.10. The observed, the predicted with MVA and the predicted with improved MVA response time in the first environment

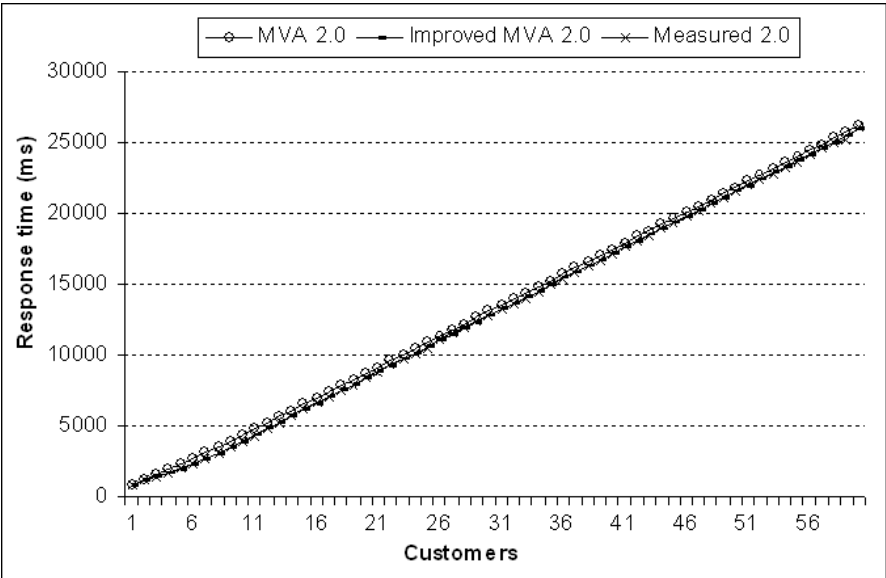


Fig. 1.11. The observed, the predicted with MVA and the predicted with improved MVA response time in the second environment

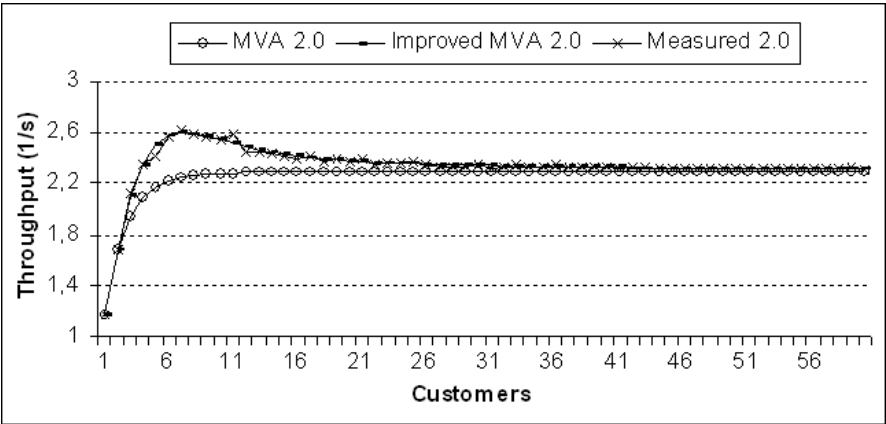


Fig. 1.12. The observed, the predicted with MVA and the predicted with improved MVA throughput in the second environment

1.5 Conclusions and future work

In this paper our proposed improved MVA algorithm has been described. Additionally, the proposed algorithm has been demonstrated in two ASP.NET environments. Namely, the MVA and improved MVA evaluation algorithms for closed queueing networks have been implemented with the help of MATLAB, the inputs have been estimated in both environments from one-one measurement, and the model was evaluated. Moreover, two measurement processes were executed in order to experimentally validate the proposed algorithm in the two environments. Our results have shown that our proposed algorithm predicts the response time and the throughput much more accurately than the MVA algorithm in both environments.

Our proposed improved MVA algorithm can be used to predict the performance of a multi-tier information system with multiple threads. This algorithm can determine the response time and throughput up to an arbitrary number of customers, only the input values of the algorithm must be estimated or measured.

In order to improve the algorithm, the limits of the four thread types in .NET thread pool, along with the global and application queue size limits must be handled. These extensions of the algorithm are parts of future work.

References

1. Bernardi S, Donatelli S, Merseguer J (2002) From UML sequence diagrams and statecharts to analysable petri net models. In: ACM International Workshop Software and Performance, pp 35–45
2. Bernardo M, Gorrieri R (1998) A tutorial on EMPA: A theory of concurrent processes with nondeterminism, priorities, probabilities and time. *J Theoretical Computer Science* 202:11–54
3. Bogárdi-Mészöly Á, Sztás Z, Levendovszky T, Charaf H (2005) Investigating factors influencing the response time in ASP.NET web applications. In: *Lecture Notes in Computer Science* 3746:223–233
4. Bogárdi-Mészöly Á, Levendovszky T, Charaf H (2006) Performance factors in ASP.NET web applications with limited queue models. In: 10th IEEE International Conference on Intelligent Engineering Systems, pp 253–257
5. Gilmore S, Hillston J (1994) The PEPA workbench: A tool to support a process algebra-based approach to performance modelling. In: 7th International Conference Modelling Techniques and Tools for Performance Evaluation, pp 353–368

6. Herzog U, Klehmet U, Mertsiotakis V, Siegle M (2000) Compositional performance modelling with the TIPPTool. *J Performance Evaluation* 39: 5–35
7. Jain R (1991) *The art of computer systems performance analysis*. John Wiley and Sons
8. King P, Pooley R (1999) Derivation of petri net performance models from UML specifications of communication software. In: 25th UK Performance Eng. Workshop
9. Kounev S, Buchmann A (2003) Performance modelling of distributed e-business applications using queuing petri nets. In: *IEEE International Symposium on Performance Analysis of Systems and Software*
10. Meier JD, Vasireddy S, Babbar A, Mackman A (2004) Improving .NET application performance and scalability (patterns & practices). Microsoft Corporation
11. Menascé DA, Almeida V (2001) *Capacity planning for web services: Metrics, models, and methods*. Prentice Hall PTR
12. Smith CU, Williams LG (2000) Building responsive and scalable web applications. In: *Computer Measurement Group Conference*, pp 127–138
13. Reiser M, Lavenberg SS (1980) Mean-value analysis of closed multichain queuing networks. *J Association for Computing Machinery* 27:313–322
14. Sopitkamol M, Menascé DA (2005) A method for evaluating the impact of software configuration parameters on e-commerce sites. In: *ACM 5th International Workshop on Software and Performance*, pp 53–64
15. Urgaonkar B (2005) *Dynamic resource management in internet hosting platforms*. Dissertation, University of Massachusetts Amherst

Chapter 2

Security enhancements for web based applications

Shivanand B. Hiremath,¹ Sameer. S. Saigaonkar²

¹Professor, National Institute of Industrial Engineering, Mumbai, INDIA

²Student, Post Graduate Diploma in Industrial Engineering, National Institute of Industrial Engineering, Mumbai, INDIA

Abstract. Web applications use inputs from HTTP requests sent by the users to determine the response. Attackers can tamper with any part of the HTTP request, including the URL, query string, headers, cookies, form fields and hidden fields and attempt to bypass the application's security mechanisms. Common input-tampering attacks include: forced browsing, command insertions, cross site scripting, buffer overflows, format string attacks, SQL injection, cookie poisoning and hidden field manipulation. In this paper, we have proposed an algorithm to detect hidden fields, the form fields and URL parameters manipulation and the algorithm is implemented in Java Server Pages (JSPTM). Security aspects are illustrated with fine remarks.

Keywords. Web Application, HTTP, URL, GET, POST, MD5, HMAC, HTTP_REFERER, JSP, MVC

2.1 Introduction

Building Web-based applications is challenging from a security perspective as HTTP transactions are connectionless, one-time transmissions. A Web transaction consists of a pair of connections - one to fetch the Web form that collects the transaction data and another to transmit the collected

data to the server [4]. For applications that lead a user through a series of input forms, the application must temporarily store field data entered on previous pages.

Developers have two choices for storing this "state" information: on the server side or on the client side. Most developers find it easy to store this "state" information on the client side, which is sent back with each transaction. State information can be stored in a browser in two ways:

- HTML form "hidden" fields
- Browser "cookies"

2.1.1 Hidden fields

The browser doesn't display hidden fields embedded in HTML and hence are called "hidden". They are an easy way for preserving state information in a web application. As the user proceeds through each input screen, you can use hidden fields to store the information already collected so it is sent back to the server as part of the next transaction.

However the hidden variable is visible in the raw HTML that the server sends to the browser. To see the hidden variables, a user just has to select "view source" from the browser menu [9].

Example: log in and change address

The following is a part of ASPTM script of "change address" application that allows a user update their mailing address.

The application displays two pages:

1. A form for mailing address information
2. A page indicating the result of the transaction

```
...  
<input type="hidden" name="userid" value="<%=userid%>">  
<input type="hidden" name="form_expires" value="<%=form_expires%>">  
<input type="submit" name="chaddr" value="Change address">  
...
```

(We are omitting a user login page, plus the script).

When we run the above script in the browser and view the HTML source code we see the values stored in "hidden" fields to store information from the login process:

```
...  
<input type="hidden" name="userid" value="nitie">  
<input type="hidden" name="form_expires" value="20070301:12:45:20">
```

...

When we fill in some information and press the submit button, a confirmation screen uses the value of the userID hidden field from the HTML form.

A hacker can use browser's "Save as" feature to save the HTML of the change address form to his computer. The complete HTML, including values in the hidden fields are saved.

Then he can open the HTML file with a text editor and change the use-rid field and save the file. And then he can alter the other hidden fields as well.

Thus he can open the file in his web browser and submit the form. As the application is trustworthy with respect to its contents of the hidden fields, any user knowledgeable enough to save and load the HTML is able to alter the user ID field and make changes freely to the addresses of any user they like.

Experienced web programmers believe that this type of tampering can be easily prevented by checking the HTTP_REFERER variable.

Most browsers send an HTTP header named "HTTP_REFERER". It contains the URL of the page the user viewed before the current one. If the hacker saves the form to his computer and resubmits it, HTTP_REFERER is blank or contains a different URL.

This is not a safe method to validate a form against tampering. Just like form fields, the value of HTTP_REFERER is set by the web browser. A user with only a little knowledge can write a script to spoof this header along with the contents of the form.

Checking HTTP_REFERER will catch trivial attempts to tamper with forms, but cannot be relied on for serious web applications [8].

2.1.2 Cookies

A "cookie" is a mechanism developed by the Netscape Corporation to make up for the stateless nature of the HTTP protocol [9]. Normally, each time a browser requests the URL of a page from a Web server the request is treated as a completely new interaction. Although this makes the Web more efficient, this stateless behavior makes it difficult to create things like shopping carts that must remember the user's actions over an extended period of time. Cookies solve this problem. A cookie is a small piece of information, often no more than a short session identifier that the HTTP server sends to the browser when the browser connects for the first time [9].

Because cookies are not part of the standard HTTP specification, only some browsers support them: currently Microsoft Internet Explorer 3.0 and higher, and Netscape Navigator 2.0 and higher [9]. Cookies are small chunks of data stored by the browser and they are of two types:

- Transient (or session) cookies: Ones without an expiry date.
- Persistent cookies: Ones with an expiry date set in the future.

Persistent cookies are stored by the web browser on disk. MS Internet Explorer, for example, stores each persistent cookie as a unique file in the user's "Application Data\cookies" folder. Netscape and Mozilla-based browser store all cookies in one file named "cookies.txt". Since they are stored in files, persistent cookies can be altered by the user using a text editor.

Transient cookies are usually kept only in memory and erased when the browser window closes, making them more difficult to alter. However, several HTTP proxies are available that allow users to view and modify transient cookies, hence they must not be considered secure.

2.2 Building blocks

If all session input from a web browser can be altered and is untrustworthy, how can a web application detect tampering?

2.2.1 Digest algorithms

Message digests, like checksums, determine a mathematical "fingerprint" of the characters in a string. The fingerprint can be compared with a saved or known good value to detect tampering [3]. Digest algorithms create a unique "signature" string for any given input data, such that it is practically impossible to alter the original data and still produce the same signature. Digests are used in SSL browser connections, Virtual Private Networks (VPNs) and Public Key Infrastructure (PKI) systems to "sign" data [8].

The same technique can be used in a web application to "sign" hidden fields. The most common digest algorithm is Message Digest 5 (MD5), but many other algorithms exist such as SHA1 and RIPEMD [8].

2.2.2 Digests and HTML forms

We can use MD5 or other digest algorithm to generate a fingerprint of hidden field data in an HTML form. We can simply concatenate the values of

all the hidden fields into a string, run it through a digest algorithm to get a fingerprint and send it out in another hidden field. When the form is submitted by the user, the fingerprint of hidden field data is generated again and compared to the original fingerprint.

If the hacker comes to know that Message Digest 5 algorithm is in operational, it is possible for him to alter the hidden fields and generate a new fingerprint from the form data.

Fortunately, Digest algorithms are designed to make it impossible to determine the contents of a message from its fingerprint. We can use this property to add a secret component to the data before generating fingerprint so that the user never gets to know. The user may be able to compute the fingerprint of the "hidden" form fields, but without knowing the secret component, he cannot compute the correct fingerprint.

By adding a secret component to a digest, we are actually constructing what cryptographers call a "message authentication code" (MAC). The standard way to do this is via a function called HMAC.

2.2.3 The HMAC standard

The standard way to use a digest for message authentication is HMAC (RFC 2104) [8]. HMAC is a generic term for an algorithm that uses a key-less hash function and a cryptographic key to produce a keyed hash function [5]. This algorithm folds the text to be authenticated with two keys and three iterations through a digest such as MD5 or SHA1.

HMAC functions are either part of or an available addition to every programming language [8]:

Perl	Digest::HMAC	http://search.cpan.org/
PHP	mhash()	http://mhash.sourceforge.net/
Java	KeyGenerator	http://java.sun.com/products/jce/
MS.Net	HMACSHA1	Dot Net Framework 1.1

Example: a tamper-resistant form

The following is a part of JSP script of "change address" application again but this time we add a MD5 HMAC "signature" to the form.

```
....
<input type="hidden" name="userid" value="<%=userid%>">
<input type="hidden" name="form_expires" value="<%=form_expires%>">
<input type="hidden" name="signature" value="<%=signature%>">
<input type="submit" name="chaddr" value="Change address">
....
```


When we run the above script in browser and view the HTML source code we see that the form has a new "hidden" field named "signature" to store a MD5 HMAC.

```
...  
<input type="hidden" name="userid" value="nitie">  
<input type="hidden" name="form_expires" value="20070301:12:45:20">  
<input type="hidden" name="signature" value="YJSG2afXQRSsvLdDXJpjFaxLLYo">  
...
```

The fingerprint was generated using the values of the three hidden fields, plus a secret key stored only on the server.

When we submit the form, the contents of the three hidden fields are combined and an HMAC using the secret key is generated. If it doesn't match the "signature" field sent with the original form, the fields have been tampered with.

Now try saving the form to the computer and edit the "userid" field again. This time the tampering is detected. This technique can also be used with web applications that store the actual data on the server and use a client-side cookie to store a session key.

2.3 Proposed approach

User's selection on html pages is stored as a form field and sent to the application as HTTP request (GET / POST). These form fields can be hidden or visible. The form fields of the URL can be easily modified by the hackers. Thus, in this manner, attacker can gain an unauthorized access to the data.

The web application can be made secured by securing hidden parameters, form input parameters and URL parameters. For each page (whether it has form element or not), we store all the hidden parameters and their possible set of values in the session as a hash table. In case of hidden parameter having more than one permissible value, all permissible values are stored in the session as a hash table. Similarly, form input parameters which are not hidden (e.g. list of values in combo box) are also stored in session as a hash table. To prevent manipulation of parameters in URL, the URL is secured by generating a key (e.g. hash value) corresponding to the URL to guarantee its integrity. Now when a request is made, the controller first checks the request for integrity of the fields and the URL and then performs necessary actions.

The following functions are implemented and used in web pages to enhance security of web based applications.

1. checkSubmit

This function is called first in each page to ensure that the web page is being called only through the request controller.

2. secureHiddenParameters

For each page (whether it has form element or not), we store all the hidden parameters and their possible set of values in the session as a hash table. In case of hidden parameter having more than one permissible value, all permissible values are stored in the session as a hash table. Then call to this function is made by passing this hash table.

3. setParameterValues

The form input parameters which are not hidden (e.g. list of values in combo box) are also stored in session as a hash table. Then call to this function is made by passing

- Name of input parameter to be secured and
- Value / list of values for that parameter.

4. allowNullValuesForParameter

If a particular form element is disabled, the value returned by selected parameter is null, hence this function is called, only if the disabled condition for that form element is true.

5. generateKeyForURL

To prevent manipulation of parameters in URL, the URL is secured by generating a key (e.g. hash value) corresponding to the URL to guarantee its integrity. Now when a request is made, the controller first checks the request for integrity of the fields and the URL and then performs necessary actions.

Each and every link generated in the web page must be secured using this function. Link generated using following must be secured.

- href
- action= (for form)
- window. (window.open or window.location)
- <iframe src=
- Request?
- Response.

If the URL is being generated in a JavaScript function by concatenating the request parameters (JavaScript variables), the part of URL except the JavaScript variables must be secured. In such cases the parameters whose value is assigned using JavaScript are not secured.

2.4 Implementation

Model-View-Controller (MVC) is the most popular and implemented architecture in industry. All of the competing web technologies like JSPTM, ASP.NetTM or PHPTM have provided an implementation of MVC. MVC organizes an interactive application into three separate modules: one for the application model with its data representation and business logic, the second for views that provide data presentation and user input, and the third for a controller to dispatch requests and control flow [2].

The following code (due to space constraints we have given pseudo code) demonstrates how web security can be enforced in J2EETM.

Let ReqController be the controller servlet of the MVC Model in the web tier and Web Security be the class which contains functions for web security.

ReqController class is responsible for processing all requests received from the users and generating necessary response to the requests.

Process Logic for detection of form fields, URL parameters, and Hidden fields is as follows:

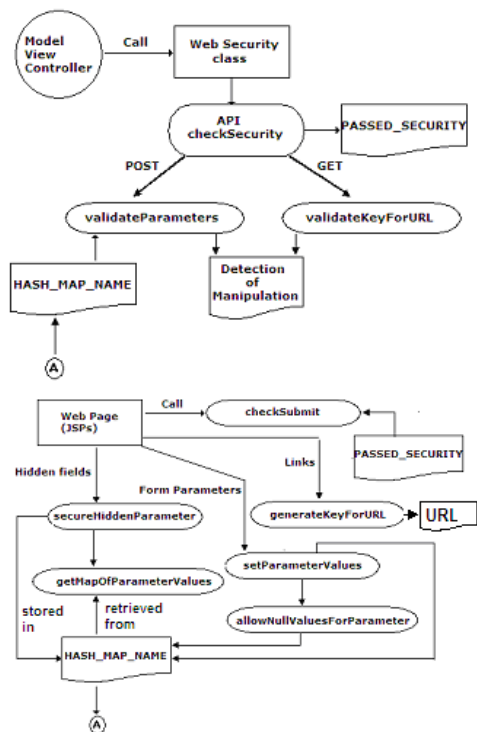


Fig. 2.1. Process logic

Pseudo code:

```
public class ReqController extends HttpServlet
{
    ...
    public void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException
    {
    ...
    WebSecurity.checkSecurity(request,response);
    ...
    }

    public void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException
    {
    ...
    WebSecuri-
ty.checkSecurity(request,response);
    ...
    }
```

```

    ...
}

```

WebSecurity class contains all the functions for enforcing Web Security.

```

// This API will be called from the ReqController //Servlet.
//It will call all security related APIs.
void checkSecurity(HttpServlet, HttpServletRequest, HttpServletResponse)
{
    Set PASSED_SECURITY = true in request.
    if (method = "POST")
    {
        Get HASH_MAP_NAME from the HttpServletRequest as w_mapName
        Call validateParameters(request,mapName);
    }
    else if(method = "GET")
    {
        Call validateKeyForURL(request,servlet);
    }
}

// This API will be called from all JSPs.
// PASSED_SECURITY attribute in Request is set
//to true if web page is being called through the
//controller.
void checkSubmit(HttpServletRequest, HttpServletResponse, ServletContext)
{
    If ( RemoteHostAddress == LocalHostAddress OR RemoteHostAddress ==
    "127.0.0.1")
    Then Ignore
    Else If request.getAttribute(PASSED_SECURITY)== null
    Then throw Exception("This jsp has bypassed Security Checks.");
}

//This API will be called from all JSPs to secure
//hidden parameters.
//The parameter a_paramValues should be as
//follows:
// e.g. "a:1,b:2,c:3:4:5,d:4:6"
void secureHiddenParameter(HttpServletRequest, HttpServletResponse, Ser-
vletContext, String paramValues)
{
    Get HASH_MAP_NAME from the HttpServletRequest as mapName
    Get HashMap corresponding to mapName from the Session as map
    map = getMapOfParameterValues(map,paramValues);
}

```

```

Set HashMap corresponding to mapName in the Session.
}

//called from secureHiddenParameter
HashMap getMapOfParameterValues(HashMap map,String paramValues)
{
Separate the parameters and their corresponding values from paramValues
Put (parameter, values) pairs in hashmap map
return map;
}

//This API will be called from all JSPs to secure
//input parameters.
void setParameterValues(HttpServletRequest,HttpServletResponse,ServletContext,String param-
Name,Object paramValues)
{
Get HASH_MAP_NAME from the HttpServletRequest as mapName
Get HashMap corresponding to mapName from the Session as map
Get ArrayList of values corresponding to paramName from the HashMap
as valueList
Add paramValues to the ArrayList valueList
Put valueList corresponding to paramName in HashMap map.
Set HashMap corresponding to mapName in the Session.
}

void allowNullValuesForParameter(HttpServletRequest, HttpServletResponse,
ServletContext, String paramName)
{
WebSecurity.setParameterValues(a_request,a_response,a_application,a_paramName,"Du-
mmy");
}

void validateParameters(HttpServletRequest,String mapName)
{
Get HashMap corresponding to mapName from the Session as map
For each parameter param in hash map
{
Get ArrayList of values corresponding to param from the HashMap as valueL-
ist
Get ArrayList of values corresponding to param from the request as reqValues
For i = 1 to no_of_values_in_request
{
For j = 1 to size_of_valueList
{

```

```

        If (w_valueList[j].equals (w_reqValues [i]))
        {
            w_matched++;
            break;
        }
    }
}
if (w_matched==0)
{
    throw Exception("The parameter has value which is not in the allowed set of
values.");
}
}
}

void validateKeyForURL(HttpServletRequest a_request,HttpServlet a_servlet)
{
    String w_secretParameterName = "key";
    String w_secretParamValue = "RandomString";
    String w_keyStr[] = a_request.getParameterValues(w_secretParameterName);
    if (w_keyStr == null || w_keyStr.length == 0)
    {
        throw Exception ("Keys Manipulated");
    }
    String w_queryStr = a_request.getQueryString();
    if (w_queryStr==null)
    {
        throw Exception ("No query String");
    }

    String w_qryStr =
w_queryStr.substring(0,w_queryStr.lastIndexOf("&"+w_secretParameterName
+"="));

    Find digest of (w_qryStr+w_secretParamValue) using "MD5" message digest
algorithm as w_NewKey.
    if(!w_NewKey.equals(w_keyStr[w_keyStr.length-1])) //Pick the last one
    {
        throw new Exception ("Query String Manipulated");
    }
}

// This API will be called from all JSPs.
String generateKeyForURL(HttpServletRequest
a_request,HttpServletResponse a_response,ServletContext a_application,String
a_url)

```

```

{
if (a_url.indexOf("?")==-1) //there is no ? in the url
{
return a_url;
}
String w_secretParameterName = "key";
String w_secretParamValue = "RandomString";
String w_temp = a_url.substring(0,a_url.indexOf("?"));
String w_params = a_url.substring(a_url.indexOf("?")+1,a_url.length());
Find digest of (w_params+w_secretParamValue) using "MD5" message digest
algorithm as w_Key.
return w_temp+"?" +w_params+"&" +w_secretParameterName+"="+ w_Key;
}

```

2.5 Conclusions

The proposed approach can detect tampering of form fields, URL parameters and hidden fields, but it is not foolproof. If GET method is used in request then the validateKeyForURL procedure is not secure enough. The weakness of this method lies in the "secret key" stored on the web server. If the server is broken into or someone manages to view the source code of the application, the method is no longer secure. However, assuming that the web server is reasonably secure from break-ins and that we change the key regularly, the approach is secure enough.

References

1. Schneier B (1996) Applied cryptography. J. Wiley
2. Singh Inderjeet, Stearns Beth, Johnson Mark, and the Enterprise Team (2002) Designing Enterprise Applications with the J2EE™ Platform. Second Edition, Addison Wesley
3. Graff Mark G, van Wyk Kenneth R (2002) Secure coding. O'Reilly & Associates
4. Smith Richard E (1999) Internet cryptography. Addison Wesley
5. Venkatramanayya S, Bishop M (2006) Introduction to computer security. Addison Wesley
6. CERT advisory CA-2000-02. <http://www.cert.org/advisories/CA-2000-02.html>
7. Hidden form field vulnerability. White papers (InfoSec Labs), <http://www.infosecclabs.com/mschff/mschff.htm>
8. Preventing HTML form tampering. <http://advosys.ca/papers/form-tampering.html>

9. World Wide Web security FAQ. <http://www.w3.org/Security/Faq/www-security-faq.html>

Chapter 3

Description and discovery of web applications in grid

Serena Pastore

INAF - Astronomical Observatory of Padova
Vicolo Osservatorio 5 – 35122 – PADOVA, ITALY

Abstract. Most of the shared distributed services on the Internet are developed as modular web applications, based on infrastructures that should provide several functionalities in order to guarantee the effective management and use of resources. Thus an adequate resources' representation is required for discovery, composition and security goals. This paper starts from the experiences gathered in the broad context of the GRID-IT project, aiming at porting in grid astrophysical applications, to report about the efforts and solutions proposed in describing a web based application resource and its relationships with discovery. Starting from a grid-based data model (GLUE schema), the descriptions involve the WSDL language, its relationships with a web service based data structure (UDDI schema) to a semantic annotation language for service capabilities (OWL-S or WSDL-S) in order to enhance the discovery task. The steps needed to process the web application in order to provide such information are provided by using existing open source software tools.

Keywords. Java web service applications, WSDL, UDDI, grid systems, distributed technologies, service discovery, semantic technologies, OWL-S, semantic web services

3.1 Introduction

Internet functions as a global network of shared distributed resources and services. Most of the services were developed as web applications by using several paradigms and programming languages (i.e. Java, PHP, etc.). The underlying distributed infrastructures that should provide main functionalities to guarantee the effective management and use of the distributed, networked resources are implemented as a grid system [1]. However, the need of specific resources' description following a detail data model that affects the management, discovery, composition and security tasks is required. This paper starts from these issues to describe experiences made in testing the solutions for working with web applications in a grid system [2]. The work has arisen from a broad framework, the GRID-IT project (<http://www.grid.it>) which aimed to establish and test the Italian grid infrastructure. In specific, the WP10 project group, conducted by INAF (<http://www.inaf.it>), had the task of studying the porting in grid astrophysical applications. The nature of such applications involves web services technologies [1] for its interoperability features, grid technologies for the exploitation of computing and storage facilities, and a secure environment (i.e. like the grid ones) to satisfy local policies. These are some of the practices that the astronomical community grouped in the IVOA alliance (<http://www.ivoa.net>) proposes in the optics of having a unified framework of resources managed by each site. The paper faces grid resource description challenges, referring to specific web application resources made up of web services deployed in the grid distributed system provided by the EGEE-II (<http://www.egee-eu.org>) infrastructure. In such a framework the lack of adequate software resource description can affect several aspects, including the discovery and use of such resources. This study provides the "application layer point of view", since an adequate description founded on the fact that the software resource is poorly described in EGEE-compliant grid system affects both the application development and the resource provider approach in the discovery task. Starting from a specific grid data model (i.e. the GLUE schema, <http://glueschema.forge.cnaf.infn.it>), the description of such resources involves the WSDL language [1], its relationship with a web service based data structure (UDDI schema, <http://www.uddi.org>), to semantic annotation languages for service capabilities to enhance discovery. The solutions proposed are based on the adoption of existing grid and web services standards through available open source software implementations that may be customized.

3.2 The project scenario and resource's provider challenges.

The EGEE-II infrastructure as the evolution of European grid projects in the last years represents an ideal grid platform for running applications. In Italy the access to the infrastructure is directed by the INFN (<http://www.infn.it>) which coordinates its use by providing a set of main grid services. The other institutions structured in Virtual Organizations (VOs) [1] contribute their resources by connecting to the grid of one or more sites, made up of physical machines which provide the logical functionalities. EGEE-like architecture [3] implemented through the gLite toolkit (<http://glite.cern.web.ch>) distinguishes logical nodes that, with customized specific software, perform the grid core services and the grid resources: the workload and data management, the information system, computing resources (CE), storage resources (SE) and processing nodes (WN). Thus, there is both a centralized management of core functionalities, needed for the entire system to work and a decentralized (VO or site-based) management of resources (through the CE, WN and SE nodes). A software resource provider is realized by the hosting environment (i.e. an application server like the Apache Tomcat, <http://tomcat.apache.org> for Java-based applications). However, their use is related to the effective integration of the software components within the grid core functionalities. In an ideal environment a consumer (i.e. human, application or agent) should be able to automatically match its requirements with the provider's opportunity which offers a network-oriented interface by which the client invokes the service. A user/client takes advantage of the grid power if there is an automatic match between the requirements and the best resources available at that particular time.

3.2.1 Grid services, grid resources and web services-based application

Grid services (i.e. discovery) provide uniform interfaces to the types of system elements (i.e. grid resources). According to grid definitions [1], resources are heterogeneous, geographically distributed and connected by a wide-area network, and accessible on-demand. However in the EGEE model [3], grid resources focus on hardware features of each VO's site. The data model adopted for resources is in fact the GLUE schema, an abstract model available for both LDAP and XML schemas, which describes the attributes and the values of each of the grid sites (Figure 3.1). It distinguishes core entities (*site* and *service*), computing resource (*cluster*, *CE*

and *subcluster*) and storage resource (*SE*, *storage area*) elements. A site collects the resources owned and managed by the same organization and it is made up of clusters (an aggregation entity that represents a complex computing resource). In this context, a software resource is described by the “*Software*” entity as a feature of the *subcluster* element, which in turn provides details of the machines (*clusters*) that offer job execution environments. The software entity describes what software packages are available in the worker nodes part of the *subcluster*. Its attributes consists essentially the name, the version and some values about the run time environment. On the other hand, the service element, which was introduced later, is not specific to the web service software and refers to the site element features. All grid resources are indexed by the grid information system (IS) [3] that is in charge of collecting and monitoring resources. Initially implemented as a LDAP directory access system (through the BDII system) [3], it is now going to be substituted with the R-GMA system (<http://www.r-gma.org>) that bases relational database store entities information into tables and uses SQL commands instead of LDAP tools for searching tasks. However, since it is based on the same data model, software discovery is made through the attributes associated to the “software” element and searching tools are thus limited to a few keywords (i.e. name and version). This becomes a problem for web service-based applications in which descriptions are rich in information and can be used in the discovery task. Software resources, like java web service applications, in fact provide a rich set of information useful for sharing, publishing, discovering and invoking them across the web in a secure way. A web service may be considered as the new breed of web applications, opposed to packaged based products. Layered upon an XML-based stack [1], it is described by specific languages that describe its capabilities and exchanges messages to be processed according to the detail specifications (i.e. SOAP protocol). A SOAP-based web service for example exposes on the network a WSDL interface that separates the abstract functionalities from the details of how and where the service is offered. According to the version 1.x of the specification, the service in the WSDL document is described (Figure 3.1) within the `<service>` element as a collection of endpoints (`<port>`). Each endpoint (port) defines a combination of binding (`<binding>`) which specifies a protocol and data format and a network address. A portType collects the operations supported that are specified in terms of input/output messages (`<message>`) and data types (`<types>`) exchanged. Unfortunately all of this information can not be used in the grid discovery task with the current core functionalities.

and publish). While the WSDL document describes the network interface, the *tModel* document provides metadata descriptions and pointers to specifications that describe the implementation. Given this provision, WSDL documents tie into the UDDI data structure in a couple of pieces and this mapping allows for the storage of WSDL information into the UDDI registry. For each WSDL document two *tModels* (one for the portType and one for the binding) are created and one *businessService* with one *bindingTemplate*. *tModel*, as an abstract structure, does not specify service or port elements. It is actually the *bindingTemplate* that is created for each unique URL access point used by the service that carries out this function. Libraries and APIs allow for finding web services through such registry (by using WSDL metadata), but it is done so in a disjointed manner in respect to standard grid mechanisms. A first integration test has been done for including such information in the LDAP grid schema [4] thanks to the RFC 4403 (<http://www.ietf.org>) that maps UDDI entities into the LDAP schema. By using the XML version of the GLUE schema, it probably will be possible to integrate the different schemas and to use the standard grid tools to find a specific WSDL or UDDI structure. Now improvement in the grid middleware has introduced the Service Discovery package [3] as a front-end to several discovery methods. This has led to further work on the UDDI solution and the adoption of richer resource descriptions. In this optics, a web services semantic description gives the possibility to enhance the automation.

3.3 A semantic description for resources

Semantics that help to describe the resources in an unambiguously computer-machine manner and are able to promote more refined search results compared to syntactic matching, is based on a slew of technologies and infrastructures based on XML, in part developed within the W3C consortium (<http://www.w3.org>). Resources are defined as classes and subclasses of objects (defined within a domain vocabulary or ontology) with relations about them (expressed with properties), and a set of inference rules. Its application in the web service area has been taken on by several projects aiming at providing languages or specific data models. Semantic web service is a research topic of the DAML program (<http://www.daml.org>) and it has its result on the OWL-S (OWL-based web service ontology) ontology language, now a W3C standard, as a technology for building semantic web services by integrating business logic with semantic web's meaningful content. The language provides a set of constructs for describing the prop-

erties and capabilities of web services. On the other hand, the W3C consortium hosts an activity that takes account of all of the semantic efforts (i.e. METEOR-S (<http://lsdis.cs.uga.edu/projects/meteor-s/>) and WSMO (<http://www.wsmo.org>) projects). It proposes the WSDL-S as a mechanism to associate semantic annotations with the WSDL-based version 2.0 (but also with WSDL 1.1) descriptions, and as a method to assess annotations to services. Finally the Semantic Web Services Framework which consists of a language and ontology has been proposed to the W3C by SWSI initiative (<http://www.swsi.org>). Even if there are a lot of initiatives in dealing with semantic web services, the work has focused on the OWL-S language since the availability of open source software to work which was developed by the Software Agent group of the Carnegie Mellon University (<http://www.cs.cmu.edu>). It is likely that the WSDL-S language will be more useful for the studied web applications, but at present there are too few tools to work with.

3.3.1 Web application description enrichment with OWL-S language and its application to discovery task

According to OWL-S, for each distinct service composing the web application there is an instance in which the class *Service*, to whom it associates three properties and thus three XML-based descriptions (*ServiceProfile*, *ServiceModel* and *ServiceGrounding*), provides all the information needed by a client or an agent to interact with (Figure 3.2).

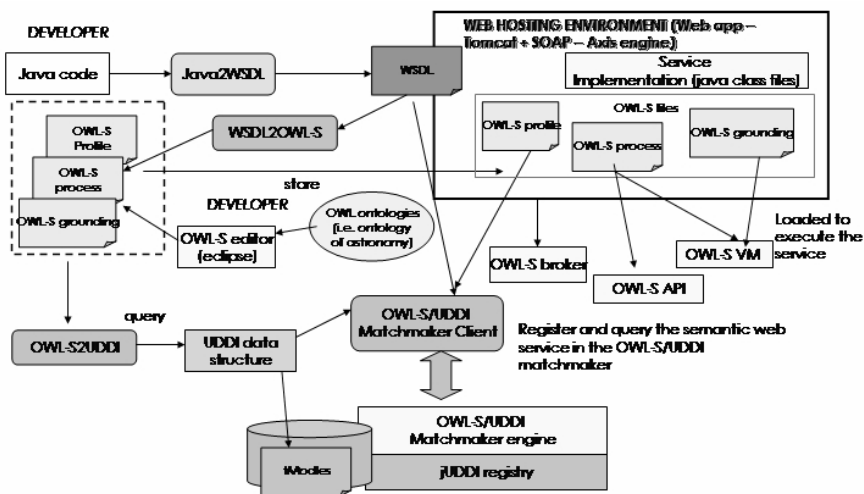


Fig. 3.2. An overall representation of all the modules and tools needed to create, deploy and use a semantic-enhanced web service application

A service can be described by at most one model; however it is associated with having only one grounding. The Profile (*presents*) defines the service's capabilities in terms of the processes performed and thus gives information to the discovery task, while the Model (*describedBy*), which describes how it is used and the Grounding (*supports*) which offers the details about transport protocols and data formats are useful to make use of once found. When the web application is semantically described, its discovery needs a customized registry enabled to parse semantic annotation when required by a client. All the steps shown in Figure 3.2 are needed to enrich the grid application description, process and use it through a discovery in the grid, which are then realized by tools and components provided by the Carnegie Mellon software. Starting on the development phase when the Java source code that is parsed by the Java2WSDL conversion, the WSDL file is exported as a network interface inside a grid machine, thus accessible by all grid consumers which are authorized to access the web service framework where the service implementation (java class files) are deployed. Moreover the UDDI registry is also deployed inside a grid node. At this point the semantic annotation of the service description is done through the WSDL2OWL-S tool which creates the needed OWL-S files. This step, however requires a further elaboration to fill out missing information and if necessary to validate it. In this phase for example, other ontologies (i.e. the astronomical ontology) are applied in order to better define the application context. All files are hosted in a web/grid accessible location. The realization of an OWL-S-enabled registry is provided by the OWL-S/UDDI matchmaker, a software that consists of an engine enhancing the capabilities of a jUDDI-based registry in order to understand and process semantic information related to service description and the client component (java library). This is then used to publish and query information on the registry. The approach takes advantage of both data model relationships and of the *tModel* concept as the way to store ontology information by realizing the mapping of OWL-S constructs with WSDL artifacts and UDDI data. In specific, OWL-S Model refers to WSDL artifacts, and OWL-S Profile to UDDI structures. The conversion between the OWL-S files and the UDDI data are made by the OWL-S2UDDI tool, while *tModels* representing OWL-S structures are stored in the relational database of the registry.

3.4 Conclusions

In the porting of grid astrophysical web application, some problems have been found in the adequate representation of such resources. The GLUE schema which is used as the reference model for EGEE-based grid system is up to now not able to fully describe these types of software resources. This has resulted in difficulty in using them in such a context. To partially overcome the problem, the solution proposed has been to introduce an UDDI registry which makes use of specific data structures to describe the WSDL-based applications that can help in managing the resources. Unfortunately the tool, installed in a grid system, is not yet usable. This will remain the case even if a middleware improvement is made, which tries to provide a front-end mechanism to different discovery backends. Moreover, this approach may take advantages of the enrichment of information brought by a semantic annotation description through the OWL-S language that, thanks to the great availability of open source tools and software, helps to further describe grid software resources and enhance its discovery. The deployed framework will help the community to learn and test grids, web services and semantic technologies in order to better describe, use and manage web applications.

References

1. Sotomayor B, Childers L (2006) Globus Toolkit 4. Elsevier
2. Volpato A, Taffoni G, Pastore S et al. (2005) Astronomical database related applications in the Grid.it project. In: ASP Conference Series, Vol. XXX, pp P1-3-5
3. EGEE JRA1 team (2004) EGEE Middleware architecture. EU deliverable DJRA1.4, EGEE-DJRA1.1-594698-v1.0
4. Pastore S (2006) The service discovery methods issue. Journal of Network and Computer Applications, Elsevier Science, to be published

Chapter 4

An approach to treat the user's preferences about personal data

Sérgio D. Zorzo, Reginaldo A. Gotardo, Paulo R. M. Cereda, Bruno Y. L. Kimura, Ricardo A. Rios, Robson E. Grande

Computer Science Department, Federal University of São Carlos
Rod. Washington Luis, km 235 - 13565-905, São Carlos – SP - BRASIL
{zorzo, reginaldo_gotardo, paulo_cereda, bruno_kimura, ricardo_rios, robson_grande}@dc.ufscar.br

Abstract. The concern with users' privacy in the Internet is an argued subject in some areas of knowledge. Mechanisms and actions exist to promote guarantees of user's privacy; the majority of them, however, present limitations in the privacy control. This paper describes a procedure for the user's privacy preferences treatment in web systems based on a contextualization model. It offers a personalization choice to the user guaranteeing that his/her privacy preferences are respected. The procedure is applied in a case study and the observed results are presented, evidencing that the privacy personalization improves the user trust in web transactions.

Keywords. Privacy, Personalization, Personal Data Protection, Web Privacy Preferences, Web

4.1 Introduction

Services personalization has a great importance in Web enterprises, because it is a kind of direct marketing which, based on users' profiles, directs contents and advertisement. According to a research conducted by

Kobsa [2], Websites that offered personalized services obtained a 47% raise in the number of new clients.

Direct marketing is substituting the traditional mass marketing strategies. However, to apply personalization, a data collection about users is necessary. This brings into consideration a discussion about privacy.

When a process of data collecting is started, the user needs to be informed about it (and agree to it) as well as be informed of how the data will be treated.

According to Teltzrow [5], 64% of Web users gave up accessing some Website, or buying something online because they didn't know how the information they gave to the system would be used.

Data collection can occur in implicit or explicit ways. Explicit collection happens when an individual sends information to a certain destiny, trusting that the information will be stored and conveniently treated. Implicit collection happens without an individual noticing it and generally consists of gathering data based on user's interactions with the system.

Implicit collection allows the creation of profiles based on users' interests, mainly inferred from his/her browsing patterns. Therefore, explicit and implicit collections differ from each other in relation to the authenticity of the information. When information is collected in an explicit mode, its authenticity is the user's responsibility. However, when the information is collected in an implicit mode, its authenticity can be questionable, since the user is unaware of the whole collecting process.

In many situations the user's confidence in web systems depends on the guarantee of privacy offered by these systems. User's suspicion about non-authorized spreading or bad use of his/her information makes him/her to give up accessing certain services.

This paper is not related to the legal restrictions applied by some legislative or normative sector. The proposed approach aims to respect the user's choice about his/her information treatment, congregating the main characteristics of legislations, norms and principles about existing data collection.

Privacy protection techniques are used to increase user's control about his/her information in Web browsing. Pseudonyms creation, use of privacy contracts and the introduction of anonymity for browsing are examples of these techniques. Each one of them approaches in different forms the user's privacy, as much for implicit collection as for explicit collection, allowing or prohibiting data acquisition for personalization application.

Based on the existing techniques, the information spreading about Websites privacy practices is the one that better enables the user to decide about his/her preferences. The user acquires knowledge about offered privacy options, deciding (or not) to subscribe to the service. However, this

technique uses static privacy policies, which make possible only one between two choices: to access or not to access a Web service.

Aiming at increasing the user control over information, this paper proposes a model inspired by the contextualization method created by Kobsa and Teltzrow [3], which implements a scheme for different possible choices for privacy treatment combined with personalization methods.

In Section 4.2 some Web privacy concepts are presented, highlighting the relevant ideas and mechanisms related to the work described in this paper. In Section 4.3, the user-controlled privacy model is described, and in Section 4.4 it is presented a case study conducting using a Website. The results and conclusions are presented in Section 4.5.

4.2 Current web privacy treatment

Web Privacy is, in a brief form, the user capacity in controlling his/her data. It can be related with users' information maintenance right in a secure way, without the possibility to identify them while browsing in Internet.

In this paper the privacy treatment will be done according to the main agent. Thus, the forms of Web privacy treatment, as such by the user, or information owner, as by the server, or information collector, will be related. To the user, tools and techniques for privacy protection are available. The server is in charge of policies and how to collect and notify the user, allowing or not the control by him/her.

There are techniques and mechanisms that offer ways to control improper information collection to the user like anonymity, pseudonyms, privacy agents and filters. Also, the privacy can be managed in the own content server for several techniques that can be synthesized in privacy policies [1, 4], negotiation protocols (P3P) [1], privacy seals [4], or privacy preferences contextualization model [3, 5].

4.3 Web privacy controlled by user

When browsing a certain Website, the user must be aware about the data collection done. The Website must show explicit, through its privacy policies, all the pertinent information about this collection. The user must read the privacy policy and, if he/she agree to it, continue his/her browsing, on the contrary, must leave the Website.

This is an approach that competes with the user browsing; therefore the user needs to observe all the privacy policies to verify which the treatment

is given to his/her privacy in the Website. This can lead to the desistance of privacy policies reading of the accessed Websites, being the user confidence based only on the Website popularity. In this form, the user's privacy is harmed, because there's no concern with the Website privacy practices, but with the image that the Website itself represents.

This paper proposes a negotiation between user and Website, in which the user defines his/her privacy profile. At the moment where the user has control of his/her privacy, defining it through his/her preferences, the Website will explicitly collect only the allowed data.

According to Kobsa, the privacy is related to the control that a user has about his/her information. In this way, transactions that provide trustworthiness to the user relating to the his/her data use, they will provide better commercial results and they will prevent failures in the business model, since the user has the fundamental role in the Website-customer relationship [2].

Web Privacy Controlled by User, presented in this paper, proposes alterations to Kobsa and Teltzrow model, treating the privacy preferences in centered way, and not contextualized. In this form, the data collection has only the global context, that can be configured according to the choices of each user.

Additionally, the concept of "privacy status" is presented, where the user can visualize his/her privacy preferences for a certain Website, as well as modifying these preferences. It is a guide who presents to the user, briefly, details about the used mechanisms for data collection. This mechanism demonstrates the continuous concern of the system in dealing with user's privacy and adjusting such treatment to his/her choices.

Through "privacy status", the user is aware about the mechanisms that are being used for data collection and how they are being used in accessed page. By the "What is this?" guide, the user obtains clear information that explains the Website intention in offering such functionalities. This guide also helps users with information about the used collection mechanisms.

Another modifier is the maintenance of the user model, which keeps his/her privacy preferences in his/her computer using Cookies, making it possible to keep an access session and offer personalized services.

The Web Privacy Controlled by User Approach modifies some of the functionalities of Kobsa and Teltzow proposal [3, 5]. Through this approach, the user obtains direct access to his/her privacy preferences and this negotiation influences in the Website behaviour. The useful benefits of the mechanisms continue presented to the user, as previously, and by "privacy status", available on the interface, the user can view and modify his/her privacy preferences.

The privacy policies, that describe the objectives and used mechanisms of collection, also present the benefits related to the each mechanism and type of data collected. In this form the user can control the data collection and he/she can verify the influence of this collection and his/her control on this.

Currently, it is predominated the use of P3P for Website privacy policy spreading. However, besides presenting low use, also it has limitations, as not to present the benefits of data collection and to request that the user takes decisions previously, without relating them to the context of use.

The use of Privacy Seals collaborates with the ratification of practices described in the Website Privacy Policy. However, the companies that supply the Privacy Seals must be trustworthy and well known by the users.

The used mechanisms by the user (privacy agents and filters, pseudonyms and anonymity) harm the offer of personalized services by Websites. Such mechanisms do not supply ways to data collection negotiation between Websites and the users, differently of the Approach proposed in this paper.

In Kobza and Teltzrow model [3] [5] it becomes more easy the reading of the privacy policies, therefore it transforms into lesser parts the reading of extensive policies. However, it becomes more onerous the developer's work, because he will must treat each aspect of the business model considering the privacy question.

It is important to say this Approach can be used with others presented mechanisms.

4.4 Case study

The Case's Study Approach was done using the Federal University of São Carlos' Graduate Course Website, in the period of November 4th to December 5th 2006. In this Website, the interested candidates in the Course get information about it, about the Institution and the Faculty. The interested parties can effect the on-line registration. The registration process needs personal information and digital documents' sending. The Approach evaluation was exactly during the period of on-line registration, because of the great access flow to the Website.

For the creation of the user's privacy preferences two mechanisms had been used: Cookies and ClickStream. Allied to them it was implemented the anonymity controlled by the Website and of user choice.

A Cookie is a set of information changed between the user's browser and the Webserver. These information are placed in an archive of text that

can be stored user's computer or only in the computer's memory, during the browsing.

ClickStream is a technique to analyze the user's browsing information in the Websites. Through the analysis of the user's clicks sequence it is possible to infer information as pages preferences or user's profile. ClickStream data can be collected using information generated by the same page, information of the Internet Service Provider (ISP) or using the servers' logs.

ClickStream acts with some data mining technique to inference of information, being also used for the generation of statistical information. When the user disables this option, his/her requests of pages won't be collected.

Every user has a sessionId for his/her identification. In this case, Session Cookies has been used and this has been excluded after finish user browsing. The information that had been recorded in the user's machine is related to his/her privacy preferences about the Website and they are only recorded with the permission of each user.

In this case, when the user accesses the Website again, it is identified the existence of a Cookie with the determined user's privacy preferences and, thus, the standard preferences are changed by the chosen ones.

One clear privacy policy was constructed, guaranteeing trustworthiness for the users. Allied to this policy, was created a page with auxiliary information that explain all the used mechanisms.

The identification anonymity was offered to the users. So, when they are using this option, the Website doesn't collect information that can identify the users, such as them IP address.

This mechanism, normally supplied for available tools to the users, was implemented in the same Website, what it demonstrates, again, the concern with the preferences of users' privacy.

In the Website Privacy Policy declaration the necessary forms for alteration of the user's privacy preferences are available. The user can observe his/her privacy status and, if necessary, modifying it. As already displayed, the Website's behavior is modified according to the alterations that have been made, differently of the other privacy proposals.

Through the guide "what is this?" the user can get information about the collection mechanisms used in the Website and the functioning of these.

In this case study it was observed that 79% of the users who had read the privacy policy – acquiring knowledge themselves of the data collection mechanisms used – had supplied their personal data to the Website for registration in the Graduate Program Course. It demonstrates that the really

interested users in the offered Course had searched information related to the security of their data and their privacy.

The Website structure contemplates information pages for visualization by the users and pages to send personal information for registration, as well as personal document copy.

This service type is highly directed to the specific public and to only one transaction object, having no generic focus, as e-commerce applications.

The measurement about privacy preferences, in this case, is related only to the interest in the Course and the registration for the selective process, differently of conventional e-commerce that offers several products and services to the users.

4.5 Conclusions

This paper contributes to the area of Personal Data Protection by proposing an approach for Web privacy preferences and discussing its implementation.

It presents quantitative data gathered during approximately one month, with constant monitoring. The results give an insight on the user's behaviour related to the Website privacy treatment. They allowed to conclude that a great amount of users read the privacy policy and were aware of the proposed approach, trusting the system and supplying personal information.

The applicability of the approach, however, depends on many factors, such as the context and the current laws, which influence its implementation.

It is a fact, however, that the common sense tells that Websites should be concerned about users' right to privacy, and respect their preferences related to the private characteristic of the information Websites collect. This can be accomplished by means of providing a personalization procedure.

The Approach to Web Privacy Preferences implementation results has proved to be an important tool for users interested in controlling their preferences, assuring their privacy when using the Web for private transactions.

References

1. Cranor LF (2002) Web privacy with P3P - The platform for privacy preferences. O'Reilly, Sebastopol

2. Kobsa A (2003) Tailoring privacy to user's needs. *Lecture Notes in Computer Science* 2109(1):303
3. Kobsa A, Teltzrow M (2004) Contextualized communication of privacy practices and personalization benefits: Impacts on users' data sharing and purchase behavior. Martin D, Serjantov A (eds) *Privacy Enhancing Technologies, Fourth International Workshop, PET 2004*, Toronto, Canada
4. Mai B, Menon N, Sarkar S (2006) Online privacy at a premium. In: XXXVI Hawaii International Conference on Systems Sciences
5. Teltzrow M, Kobsa A (2004) Communication of privacy and personalization in e-business. In: *Proceedings of the Workshop "WHOLEs: A Multiple View of Individual Privacy in a Networked World"*, Stockholm, Sweden

Chapter 5

Ssort: Peak analysis for efficient string sorting

Sunil Mallya, Kavi Mahesh

Department of Computer Science and Engineering, PES Institute of Technology, 100 Feet Ring Road, Bsk 3rd stage, Bangalore 560085, INDIA
mallya16@gmail.com, drkavimahesh@gmail.com

Abstract. With the growth of the Internet, processing large amounts of textual data has become a common requirement in practical applications. This paper presents an efficient algorithm, Ssort that is specialized for sorting large numbers of strings. It is an improvement over Quicksort and its variants such as Meansort. It has an excellent average case behavior. Moreover, its worst case is highly unlikely to occur in practice.

Keywords. String sorting, Partition ratio, Artificial pivot induction, Peak-analysis

5.1 Introduction

In today's world, due to the advent of the internet and large-scale text databases, it has become increasingly important to have a good text processing toolkit. String sorting is an essential part of such a text processing toolkit. Unfortunately, currently available sorting algorithms are not ideal for sorting long lists of strings and not much work has been done in this domain. Most sorting algorithms available for sorting numbers are extended to sort strings with a suboptimal performance and thus there is a need for a specialized algorithm for sorting strings.

Divide and conquer techniques for sorting algorithms have been known for their superiority in sorting by comparison. Of all the algorithms of this

paradigm, Quicksort by Hoare [2] is the most popular algorithm due to its efficiency parameters. One of the drawbacks of Quicksort is the worst case behavior of $O(n^2)$ comparisons, in the standard version of Quicksort. This case will occur when the file is sorted or reverse sorted. The Meansort algorithm by Motzkin [1], which is a modification of Quicksort algorithm calculates the arithmetic mean and then partitions the array. This improves the Quicksort algorithm because arithmetic mean provides for a better pivot resulting in two almost equal partitions. Meansort has not been extended to strings. Meansort algorithm cannot be applied directly to string sorting without significant changes. Meansort will fail to achieve good results when the input is concentrated in a particular area. For example,

Eg1: If we take the input to contain 10 strings starting with the letter 'a': a1, a2...a10 and 100 strings starting with the letter 'z': z1, z2...z100,

The arithmetic mean calculated would be 'x', resulting in a left partition having 10 elements and a right partition having 100 elements. This is highly undesirable because such repeated bad partitions can result in $O(n^2)$ growth. Drawbacks of Meansort algorithm are overcome in our algorithm by a method called Artificial Pivot Induction in which the mean is shifted by a small constant factor, the API factor. This method follows the principle of Locality of Reference and tries to find a better pivot in the vicinity of the current mean. This will work for some cases, but for the example described earlier we need a better method to tackle the problem. This is accomplished by our algorithm through a method called Peak-Analysis. In Peak-Analysis, as illustrated in more detail below, we choose strings that belong to the maximum character count or 75% of the maximum count and the arithmetic mean of this reduced set is taken. This ensures that the mean is influenced only by input strings that begin with the character having the maximum count, thus ensuring a better partition.

The Algorithm described in this paper has three main parts, namely, Mean Partitioning, Artificial Pivot Induction and Peak-Analysis. The first part of the algorithm partitions the input based on the arithmetic mean of some prefix of the strings to be sorted. Artificial Pivot Induction tries to induct a pivot artificially when the first part of the algorithm fails to achieve a good partition. The technique of Peak-Analysis computes a better pivot string when the initial part of the algorithm fails to get a good pivot by finding the mean. The algorithm tries to get a partition ratio around 1, which is defined to be the ratio of number of elements in the left sub-array to the number of elements in the right sub-array, i.e., the array is divided into equal sub-arrays. Finally the overall algorithm achieves an efficiency of $\Omega(n \log_2 n)$ for the best and $\theta(\square n \log_2 n) \square \square$ average cases. As shown later, this algorithm has a lesser number of partitioning steps. This

algorithm is unlikely to encounter an $O(n^2)$ growth for practical inputs. The superiority of this algorithm is that it has the same time complexity and space complexity for a given data set and does not depend on the ordering of the input elements.

5.2 Algorithm description

During the first pass, the mean of the first characters of all the strings is calculated. This is our initial compare string based on which we partition the strings. We keep a String Count Pointer which keeps the index of character under comparison. Three cases can occur: the partition is good and is accepted, one of the partitions is empty thus indicating that all the strings have the same character as the mean that was used for comparison, or an uneven (or skewed) partition indicating that there is concentration of input in a certain area as illustrated in the example with a's and z's shown earlier. In the second case, we attempt to increment the String Count Pointer by 1 character and repeat the partitioning procedure. When the third case occurs, we can artificially induct a pivot and then check if this has achieved a better partition ratio. If yes, we partition and sort, else Peak-Analysis is done to break the input concentration that is described as a "Peak". Artificial Pivot Induction works well for cases such as the one in the following example:

Eg2: Consider 10 strings a1, a2, a3, c1, c2, c3, f1, g1, z1, z2.

The mean calculated would be 'g' and results in a partition ratio of 4:1. Now, because the left array is heavy, we can reduce the mean artificially by an API factor of 2 and get the new mean to be 'e'. The new partition ratio is 3:2, which is an improvement over the earlier method. After sufficient experiments and analysis 2 was found to be a good API factor and catered well to the principle of Locality of Reference. Artificial Pivot Induction works well when the mean is affected by a few larger valued strings or very low valued strings, as was the case in example Eg2. But Artificial Pivot Induction fails when there is a Peak. In a Peak, the Peak elements are the ones that begin with a character that has the maximum count (or has 75% or more of the maximum count). In Peak-Analysis, a mean compare string that is the pivot is computed based only on these Peak elements. This ensures that the mean compare string will partition the elements with an acceptable partition ratio since the new mean that is calculated is not influenced by other strings that are not present in the Peak. The Peak-analysis is done only if the sub-array has greater than 1000 elements since Peak -Analysis has an overhead which makes it profitable only for a

large number of elements. The sort is finished when there are no more sub-arrays. To better understand Peak-Analysis, let us consider an example:

Eg3: consider 1000 strings $a_1, z_1, z_2 \dots z_{999}$

The mean calculated would be 'y' which would achieve a partition ratio of 1:999 and Artificial Pivot Induction would not help either. In such cases, Peak-Analysis will choose 999 strings ($z_1, z_2 \dots z_{999}$) that constitute the Peak and the new mean string calculated based only on these strings will be 'z540', thus achieving a good partition ratio. The cost of the Peak-Analysis has been determined to be $O(n)$. Most of the parameters such as the Artificial Pivot Induction factor and the maximum count for peak element selection can be set according to the nature of input dataset.

5.3 Algorithm

This section provides the detailed Ssort algorithm in the form pseudo code. Since the algorithm is fairly complex, it is essential to include the entire algorithm in the paper.

```

BeginProcedure avg_sort (string_array,n,str_count_pointer,compare_string)
/* input_array is array of pointers to the string array to be sorted */
if (n=1)
    input_array[counter] ← string_array[0]
    counter ← count +1
    return
endif
for(i ← 0 to n)
    alpha_count[string_array [i][0]] ← alpha_count[string_array [i][0]] +1
/* determine the count of first characters of the strings */
    sum ← sum +string_array [i][str_count_pointer]
/* sum of characters pointed by current string count pointer */
endfor
avg = sum / n
compare_string [str_count_pointer] ← avg
compare_string [str_count_pointer +1] ← '\0'
do {
    for (i ← 0 to n)
        if (strcmp(string_array[i],compare_string,str_count_pointer) <=0)
            if ( !strcmp(string_array[ i ],compare_string,str_count_pointer) )
                if(!strcmp(string_array[ i ], compare_string))
                    dupe_count ← dupe_count +1
            /* this is done to identify duplicate strings and to make the sort stable by restoring
            the position of the duplicate elements */

```

```

        endif
    endif

    if(dupe_count < 2)
        left_array[l_c] ← string_array[i]
        l_c ← l_c + 1
    else
        right_array[r_c] ← string_array[i]
        r_c ← r_c + 1
    endif
else
    right_array[r_c] ← string_array[i]
    r_c ← r_c + 1
endif
endfor

restore_flag ← 0
if (r_c = 0)
    if (api_flag = 1)
        compare_string[str_count_pointer]
        ← compare_string[str_count_pointer] - api_factor
        restore_flag ← 1
        /* this is used when artificial pivot induction results in a new mean which is
           beyond the boundary of strings under comparison. */
    endif
else
    str_count_pointer ← str_count_pointer + 1
    no_partiton_flag ← 1
    /* all strings have the same characters pointed by the current String count pointer
       as the mean calculated, so increment the pointer and partition based on the
       next character. So skip the artificial pivot induction and peak analysis code */
endif

if (l_c = 0 and api_flag = 1)
    compare_string[str_count_pointer]
    ← compare_string[str_count_pointer] + api_factor
    restore_flag ← 1
endif
if(no_partiton_flag != 1)
    if(restore_flag != 1)

        partition_ratio ← l_c / (float)r_c
        partition_count ← partition_count + 1
        if (partition_ratio > 2 and partition_count < 2 and n > 100)
            /* Artificial Pivot Induction:
               left array has more elements so decrease the mean */
            compare_string[str_count_pointer]

```

```

                                ←compare_string[str_count_pointer ] -api_factor
    api_flag ←1
    endif

    else if (partition_ratio<0.3 and partition_count <2 and n>100)
/* Artificial Pivot Induction:
    left array has lesser elements so increase the mean */
    compare_string[str_count_pointer ]
                                ←compare_string[str_count_pointer ] +api_factor
    api_flag ←1
    else
        api_flag ←0
    endif
endif
} while ((api_flag =1 and partition_count <2) or restore_flag =1)

if ( partition_count >=2 and api_flag=0 and n>1000)
/* Peak-Analysis */
big ←alpha_count[33]
big_index ←33
for(index ←34 to 123)
    if(big <=alpha_count[index])
        big ←alpha_count[index]
        big_index ←index
    endif
endfor

/* determine the largest count and the corresponding strings that begin with it
and put them in a new array called the peak_array */

index ←big_index
while (index <123 && alpha_count[index] >= (0.75*big))
    alpha_array[index] ←1
    index ← index +1
endwhile
index ←big_index
while(index >33 && alpha_count[index] >=(0.75*big))
    alpha_array[index] ←1
    index ←index -1
endwhile
for(index ←33 to 123)
    if(alpha_array[index] =1)
        peak_array[peak_index] ←index
        peak_index ←peak_index +1
    endif
endfor
for(peak_index2 ←0 to n)
    for(peak_index ←0 to peak_index)
if(string_array[peak_index2][0] =peak_array[peak_index1])

```

```

        peak_string_array[peak_n] ← string_array[peak_index2]
        peak_n ← peak_n + 1
    endfor
endfor
call procedure peak_avg_find(peak_string_array, peak_n, peak_compare_string)
wcs ← -1

do {
    l_c ← 0; r_c ← 0; wcs ← wcs + 1;
    if(wcs != 0)
temp_len ← temp_len + 1
/* peak compare string can result in r_c = 0, to avoid this anomaly we partition
   based on each character if wcs is set to any value other than '0' */
    endif
    for (i ← 0 to n)
        if(wcs = 0)
            temp_len ← strlen(string_array[i])
        endif
        c_count ← c_count + 1
        if ( strcmp(arr[i],
peak_compare_string, temp_len) <= 0)
            left_array[l_c] ← string_array[i]
            l_c ← l_c + 1
        else
            right_array[r_c] ← string_array[i]
            r_c ← r_c + 1
        endif
    endfor
    if(wcs_flag = 0)
        temp_len ← 0; wcs_flag ← 1
    endif

} while(r_c = 0 or l_c = 0)

endif
endif /* for no_partition flag */
if(l_c != 0) avg_sort(left_array, l_c, str_count_pointer, compare_string)
if(r_c != 0) avg_sort(right_array, r_c, str_count_pointer, compare_string)
endprocedure avg_sort

Begin procedure
peak_avg_find (peak_string_array [], [40], peak_n, peak_compare_string)
    for ( x ← 0 to max_str_len )
        peak_string_sum[x] ← peak_string_avg[x] ← 0
    endfor
    for ( x ← 0 to < peak_n )

```

```
        for(y ← 0 to peak_string_array[x][y] != '\0')
peak_string_sum[y] ← peak_string_sum[y] + peak_string_array[x][y]
        endfor
    endfor
    for(x=0 to y)
        peak_string_avg[x] ← peak_string_sum[x] / peak_n
        if(peak_string_avg[x] < '!' and peak_string_avg[x] != '\0')
peak_string_avg[x] ← 33
/* ignore null character */
        endif
        if(peak_string_avg[x] > 'z')
peak_string_avg[x] ← 122
        endif
    endfor
peak_string_avg[x] ← '\0'
strcpy(peak_compare_string, peak_string_avg)
return peak_compare_string

endprocedure peak_avg_find
```

5.4 Result analysis

The WordNet [5] lexical database [4] was used as the dataset to be sorted. It had a set of 116,938 Strings which was replicated as lowercase and uppercase words to expand the dataset to 233,876 strings. Different test datasets were generated from the original dataset by randomizing the data. Randomizing was done by repeatedly applying a card shuffling algorithm. The standard version of Quicksort was used where in the first element of the sub-array was chosen as the pivot. As the data in Table 5.1 indicates, fewer number of partition steps were required by Ssort to sort the input data. Quicksort only showed improvements for totally random data. It is evident from Table 5.2 that the number of bad partitions, which are partitions that had a ratio greater than 2 or lesser than 0.5, were found to be far lesser than in the case of Quicksort. In the best case for Quicksort, the number of bad partitions was still found to be twice as that of Ssort. The largest bad partition was found to be 121 times than that of Ssort. Even in the best case of Quicksort it was 1.3 times than that in Ssort as indicated in the Table 5.2. Time analysis showed that for a randomization factor 1 that is for highly shuffled data the performance of quicksort was better. But for any randomization factor lesser than 0.75, Ssort was found to be far better. Quicksort varied from 197 seconds to 0.41 seconds to sort from the worst case to the best case, whereas Ssort was consistent and did not depend on

the order of input elements and was measured to sort in 1.01 seconds in all cases. Time was measured on a Intel Core Duo T2050, 1GB RAM machine running on Fedora Core 5 Linux Operating System. This worst case behavior of Quicksort makes it unreliable to be used for real life applications of internet or text processing where there is a time constraint. It is undesirable for the user to wait for long periods (e.g., for a web page to refresh) merely because the data involved was a worst case for Quicksort.

Table 5.1. Comparison of Ssort and Quicksort

Randomizing factor*	Total partitions		Time in seconds	
	Quicksort	Ssort	Quicksort	Ssort
0	378343	233696	196.99	1.01
0.1	331633	233696	80.33	1.01
0.5	233876	233696	3.30	1.01
1	233876	233696	0.41	1.01

* Randomizing factor 0 indicates sorted input.

Table 5.2. Partition Analysis Ssort and Quicksort

Randomizing factor*	Number of bad partitions		Largest bad partition	
	Quicksort	Ssort	Quicksort	Ssort
0	263343	49145	108557	897
0.1	263343	49145	108557	897
0.5	109084	49145	1169	897
1	76714	49145	1169	897

* Randomizing factor 0 indicates sorted input.

It may be noted that fully randomized data is unlikely to occur in practical cases of string sorting. Because of the way in which large collections of strings are put together in any practical application, it is highly likely that there will be pockets of locally sorted data (albeit partially sorted), thereby indicating that the worst case for Ssort of fully randomly ordered strings where Quicksort is much better than Ssort is unlikely to occur in real applications.

As can be seen from the Table 5.2, the largest bad partition in Ssort was still of size 897. This is because Peak-Analysis is not done for sub-arrays having less than 1000 strings. Due to the overhead of Peak-Analysis it becomes desirable to use it only for sub-arrays which have greater than 1000 strings. The worst case for Ssort will occur where the algorithm goes to peak-analysis every time as it recursively partitions the dataset. But since peak-analysis and partitioning on mean are mutually exclusive in the algorithm, the algorithm is highly unlikely to suffer from $O(n^2)$ growth.

5.5 Further work

Though for practical input data we claim that the worst case for Ssort is very unlikely to occur, Peak-Analysis can fail when there is a “Peak” within a Peak as illustrated next. Ignoring the fact that Peak-Analysis is done for partitions with more than 1000 elements for the present, let us consider that Peak-Analysis is done for a Peak array whose contents are aa, aza, azza, azzza, azzzza. The mean compare string calculated would be ‘au’ which would result in a partition ratio of 1:4. Again for the next iteration, the mean compare string would be ‘azu’ and the partition ratio would be 1:3. This keeps recurring, resulting in a $O(n^2)$ growth. But this kind of input is highly unlikely to be encountered in practical cases. This is the worst case for Ssort and further work on Ssort can be done to achieve a better Peak-Analysis algorithm so that such worst case occurrences can be eliminated.

5.6 Conclusion

Experimental results show that Ssort is an improvement over Quicksort in terms of achieving better partitions. It also is consistent, that is, its performance is not dependent on the input ordering, a behavior which is not seen in Quicksort. Quicksort has a varied performance and its time efficiency can degrade to $O(n^2)$. The occurrence of an $O(n^2)$ worst case for Ssort is very remote because the Artificial Pivot Induction and Peak-Analysis parts of the algorithm seldom allow for a bad partition. It is a stable sort and is best suited for real time processing where the sorting has to be done in a fixed amount of time and any $O(n^2)$ growths cannot be tolerated. This is especially significant in today’s world where processing textual data in large quantities is an essential requirement in many practical applications.

References

1. Motzkin D (1983) Meansort. Communications of the ACM 26(4):250-251
2. Hoare CAR (1961) Quicksort (Algorithm 64). Communications of the ACM, 4(7):321-322
3. Morato Jorge et al. (2004) WordNet applications. In: GWC 2004, pp 270-278
4. Fellbaum C (1999) WordNet: an electronic lexical database. Cambridge, MIT Press
5. <http://wordnet.princeton.edu>

Chapter 6

Parametric studies for the AEC domain using IntelGrid platform

Matevž Dolenc, Robert Klinec

Faculty of Civil and Geodetic Engineering, University of Ljubljana, Jamova 2, 1000 Ljubljana, Slovenia, {mdolenc,rklinec}@itc.fgg.uni-lj.si

Abstract. Parametric studies play a vital role in science and engineering in general. They usually consist of a number of repetitive, independent calculations that are therefore easily parallelized. Scientists and engineers involved in this type of work need a computing environment that delivers large amounts of computational power over a long period of time. Such environment is provided by the IntelGrid project testbed. The paper introduces the IntelGrid platform and provides a detailed description of the demonstrated IDA parametric studies and its execution on the IntelGrid testbed as well as benchmark analysis of the benefits provided by the deployed Condor high-throughput computing system.

Keywords. Parametric studies, Civil engineering, High-throughput computing, Grid computing, Web portal, IntelGrid

6.1 Introduction

Parametric studies in science and engineering usually consist of a number of repetitive, independent calculations that are therefore easily parallelized. Scientists and engineers involved in this type of work need a computing environment that delivers large amounts of computational power over a long period of time. Such environment is called a High-Throughput Computing (HTC) environment. The EU project IntelGrid (2004-2007) [4]

combined and extended the state-of-the-art research and technologies in the areas of semantic interoperability, virtual organisations and grid technology to provide diverse engineering industries with a platform prototype with flexible, secure, robust, interoperable, pay-per-demand access to information, communication and processing infrastructure. As one of the important requirements for the IntelliGrid platform recognised by the end-users is the ability to perform complex parametric studies.

6.2 IntelliGrid architecture components

IntelliGrid delivers a generic grid-based integration and web semantic-based interoperability platform for creating and managing networked virtual organisations. The following common characteristics can be defined for all IntelliGrid services as main components that can be deployed either at some workstation or on the grid: (1) services are modular components that can be semantically described, registered, discovered and finally used by clients, (2) services can participate in specific business workflows, where the order in which they are sent and received affects the outcome of the operations by a service, (3) services may be completely self-contained, or they may depend on the availability of other services, (4) services are able to advertise details such as their capabilities, interfaces and supported communication protocols according to pre-defined concepts and ontologies, and (5) all capabilities provided by services as well as communication and data channels among them and clients are protected by security mechanisms.

There are three main kinds of components on the IntelliGrid platform. Firstly, there are business applications that are the consumers of other components. Most services are accessed through portal, but access through other desktop software is possible as well. Secondly, IntelliGrid platform identifies services that comply with WSRF [12] services or pure web services extended with GSI [6], and talk to each other as well as to the client applications. They can be further divided into (a) interoperability services, (b) business or domain specific services, and (c) middleware services that offer traditional grid middleware functionality. All these resources are available through a well defined secure protocols and grid-enabled generic core services for remote data access.

As the detailed discussion of the IntelliGrid architecture is out-of-scope of this paper, only a short overview of the components enabling HTC environment is presented (see Figure 6.1):

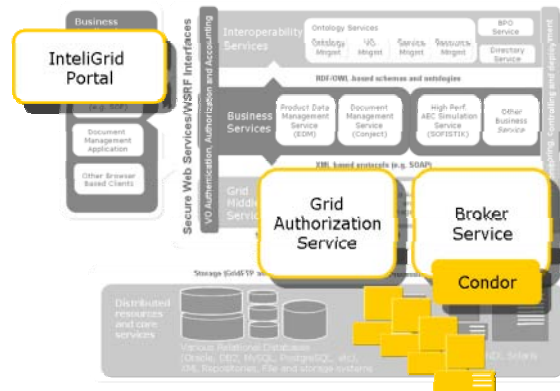


Fig. 6.1. IntelliGrid architecture components supporting HTC environment

The screenshot shows the 'OpenShell Client 1.0' window with a 'Tasks list' table and a 'Project' panel on the right.

No.	Input File	Job ID	Job Status	Remove	Submit	Cancel
1	C:\Users\Client 3\Documents\IntelliGrid\11.0\1	IntelliGrid-11.0-1	None (0)	Yes	Yes	Yes
2	C:\Users\Client 3\Documents\IntelliGrid\11.0\1	IntelliGrid-11.0-2	None (0)	Yes	Yes	Yes
3	C:\Users\Client 3\Documents\IntelliGrid\11.0\1	IntelliGrid-11.0-3	None (0)	Yes	Yes	Yes
4	C:\Users\Client 3\Documents\IntelliGrid\11.0\1	IntelliGrid-11.0-4	None (0)	Yes	Yes	Yes
5	C:\Users\Client 3\Documents\IntelliGrid\11.0\1	IntelliGrid-11.0-5	None (0)	Yes	Yes	Yes
6	C:\Users\Client 3\Documents\IntelliGrid\11.0\1	IntelliGrid-11.0-6	None (0)	Yes	Yes	Yes
7	C:\Users\Client 3\Documents\IntelliGrid\11.0\1	IntelliGrid-11.0-7	None (0)	Yes	Yes	Yes
8	C:\Users\Client 3\Documents\IntelliGrid\11.0\1	IntelliGrid-11.0-8	None (0)	Yes	Yes	Yes
9	C:\Users\Client 3\Documents\IntelliGrid\11.0\1	IntelliGrid-11.0-9	None (0)	Yes	Yes	Yes
10	C:\Users\Client 3\Documents\IntelliGrid\11.0\1	IntelliGrid-11.0-10	None (0)	Yes	Yes	Yes
11	C:\Users\Client 3\Documents\IntelliGrid\11.0\1	IntelliGrid-11.0-11	None (0)	Yes	Yes	Yes
12	C:\Users\Client 3\Documents\IntelliGrid\11.0\1	IntelliGrid-11.0-12	None (0)	Yes	Yes	Yes

The 'Project' panel on the right includes buttons for 'Open...', 'Save as...', 'Add Files...', 'Add Directory...', 'Close All', 'Submit', 'Cancel Job', 'Get Results...', 'Reset', 'Status', 'Configuration', 'Exit...', 'Misc', 'About...', and 'Quit'.

Fig. 6.2. Condor pool job monitor

- *Portal framework.* The IntelliGrid Portal is based on the GridSphere portal framework [8] which provides an open source portlet based Web portal. GridSphere enables developers to quickly develop and package third-party portlet web applications that can be run and administered within the GridSphere portlet container.
- *Grid authorization service.* Build on GAS [5] that has been designed as a trusted single logical point for defining security policies and enables the implementation of various advanced virtual organisation security scenarios.
- *Resource management.* It is build on a GRMS service [7] which allows developers to build and deploy resource management systems for large scale distributed computing infrastructures.

- *Job management.* Condor [2] is a batch queuing system for managing compute-intensive jobs. It performs it by providing a HTC environment that efficiently harnesses and makes the best use of all available resources (Figure 6.2) while providing a high throughput for jobs.

6.3 Use case: parametric studies in civil engineering

Recent efforts in civil engineering and especially earthquake engineering science are aimed towards moving the design practice away from the traditional single limit state deterministic techniques. Instead, multiple performance objectives in terms of probabilities of different socio-economic decision variables being exceeded in a given earthquake hazard environment are considered. Such decision variables relate the performance of structure during an earthquake event with decision making and involve the cost of repair, indirect cost of down time, etc.

The probabilistic performance assessment of structure is a difficult and time consuming task, especially due to complex non-linear dynamic analyses, which are performed for a variety of ground motion records scaled to a number of intensities, and due to many uncertainties involved in the nature of the problem. Because of the complex model it is practical to disaggregate the probabilistic performance assessment of a structure into independent models like loss model, capacity model, demand model and seismic hazard model. In this general probabilistic framework the relation between relevant parameters is determined with the so called IDA analysis [11], which requires a large number of non-linear dynamic analyses. Each structural model has to be subjected to several ground motion records, scaled to multiple levels of intensity in order to obtain enough data about the behaviour of the structure. It is obvious, that the number of analyses and therefore the required computer time are significant.

The bridge (see Figure 6.3) was analysed using the IDA analysis to determine the relation between the engineering demand parameter and seismic intensity measure. OpenSees [9] computer code was used to model the bridge. Several steps were taken during the analysis:

1. *Numerical modelling.* The numerical modelling of the structure that was the subject of investigation was performed. Since near-collapse region was investigated, all possible failure mechanisms had to be accurately modelled.
2. *Earthquake records.* Selection of earthquake records (this is typically selected from an Internet database such as the ESD [1]) was chosen. Due to the characteristics of the database's user interface, the

download of earthquake records cannot be automated after the selection has been made. The ground motions are normalised.

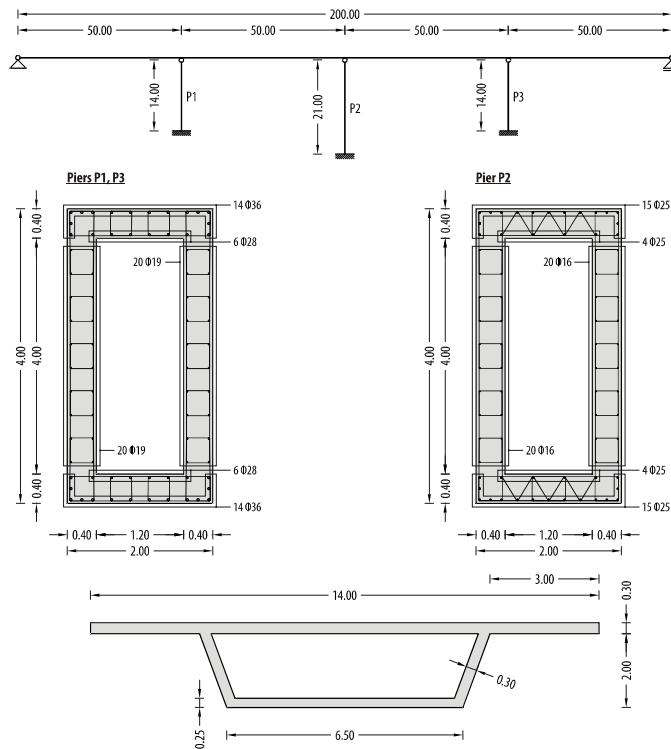


Fig. 6.3. Test structure characteristics: the box section deck (pinned at the abutments), the concrete piers (rectangular hollow cross section, designed according to the Eurocode standards), the piers (side piers - 14 m high, central pier - 21 m high)

3. *Creation of job files.* Job files for Condor, automated with Matlab, were prepared. The performance of the structure was evaluated using IDA analysis, a dynamic equivalent of the static pushover analysis. Following the procedure, the structural model was subjected to several ground motion records, each scaled to multiple levels of intensity. A nonlinear dynamic analysis was performed at each step. Several engineering demand parameters were monitored.
4. *Running the analyses.* This step was performed five times with a different number of available computing resources to determine the benefits of using the HTC environment for performing parametric studies. Figure 6.4 summarises calculation times for performing the analyses and clearly shows the advantages of parallelization of parametric studies.

5. *Post analysis.* After all load cases had been evaluated, the acquired data was analysed. The goal was to provide probability of exceeding a chosen performance level, such as the collapse prevention damage state. Cornell's approach [3] to assess the mean annual probability of exceeding a limit state was used in this example.

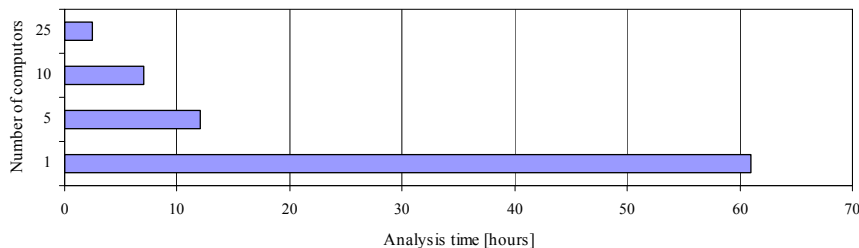


Fig. 6.4. IDA analysis computational times compared to the number of computers used

6.4 Conclusion

Parametric studies play a vital role in science and engineering in general. The IntelliGrid project, while focusing on providing data access, integration and management, lifetime project/product management and other services for engineering domains, also provides a high-performance and high-throughput environment based grid technology.

A brief introduction to the IntelliGrid architecture was presented with an emphasis on essential components enabling HTC environment. The Condor batch queuing system is used within the IntelliGrid platform that manages a pool of available non-dedicated computational resources.

To illustrate the benefits of using high-throughput computing environment for performing parametric studies an exemplary IDA analysis of a bridge was performed. The benchmarking analysis established a nearly linear speedup of the performed IDA analysis using the HTC environment with the available computing resources.

References

1. Ambraseys N, Smit P, Sigbjornsson R, Suhadolc P, Margaris B (2002) Internet site for European Strong-Motion Data European Commission. Research Directorate General, Environment and Climate Programme

2. Condor Project. www.cs.wisc.edu/condor/
3. Cornell CA, Fatemeh J, Hamburger RO, Foutch DA (2002) Probabilistic basis for 2000 SAC federal emergency management agency steel moment frame guidelines. *Journal of Structural Engineering* 128 (4):526-533
4. Dolenc M, Turk Z, Katranuschkov P, Kurowski K, Hannus M (2005) Towards grid enabled engineering collaboration environment. In: Topping BHV (Ed) *Proceedings of the Tenth International Conference on Civil, Structural and Environmental Engineering Computing*, Civil-Comp Press
5. Grid authorization service (GAS). www.gridlab.org/gas
6. Grid security infrastructure (GSI). www.globus.org/security/overview.html
7. GridLab resource management system. <http://www.gridlab.org/grms>
8. GridSphere portal framework. <http://www.gridsphere.org>
9. Mazzoni S, McKenna F, Scott MH, Fenves GL, Jeremic B (2006) Open system for earthquake engineering simulation (OpenSees) [Command Language Manual]. Berkeley, University of California, PEER
10. Pinto AV (1996) Pseudodynamic and shaking table tests on R. C. Bridges (Report No. 5). Ispra: The European Laboratory for Structural Assessment
11. Vamavatsikos D, Cornell CA (2002) Incremental Dynamic Analysis. *Earthquake Engineering and Structural Dynamics* 31(3):491-514
12. WS-Resource Framework (WSRF). www.oasis-open.org/

Chapter 7

An auction-based resource allocation strategy with a proportional share model for economic-based grid systems

Kuan-Chou Lai,¹ Chao-Chin Wu,² Jhen-Wei Huang¹

¹Department of Computer and Information Science, National Taichung University, Taiwan, R.O.C.

²Department of Computer Science and Information Engineering, National Changhua University of Education, Taiwan, R.O.C.

Abstract. Due to the complexity caused by distributed computing and resource heterogeneity, the resource management and application scheduling in grid systems are complex. This paper proposes a flexible auction-based resource allocation strategy with a proportional share model for economic-based Grid systems. Instead of maximizing resource utility, this proposed strategy aims to complete jobs by meeting their deadline and budget requirements with the lowest resource costs. The experimental results show that the proposed strategy could complete jobs before the deadline with lower budgets.

Keywords. Grid, economic model, action model, proportional share

7.1 Introduction

Grid systems have been adopted as the next generation platform for many scientific applications. The Grid system consists of geographically distributed computer systems which make use of the internet to connect dynamic, shared resources and services [4]. In order to share, integrate and man-

age these resources and services, Grid systems adopt resource allocation and management strategies to enhance the system throughput and utilization. In Grid systems, the producers (i.e., resource owners) and consumers (i.e., resource users) have different purposes and targets respectively from system-centric and user-centric viewpoints. System-centric approaches aim to optimize system performance; however, user-centric approaches, which are common to conform to the economic principle, aim to maximize utility based on users' QoS requirements [2].

In an economic-based Grid system, resource allocation and scheduling decisions are made according to the user requirements at runtime. The economic model charges users for shared resources and services that they consume basing on the costs which are defined for running applications. A user adjusts the budget to acquire resources and services to compete with other users by his demand.

In this study, we integrate the proportional share model with the first-price sealed-bid auction model [2] to develop a new resource allocation strategy which aims to complete jobs by meeting their deadline and budget requirements with the lowest resource costs. The experimental results show that the proposed strategy could complete jobs before the deadline with lower budgets. The rest of this paper is organized as follows. Section 7.2 introduces related works. Section 7.3 presents the proposed algorithm. Experimental results and performance analyses are provided in Section 7.4. Concluding remarks are then offered.

7.2 Related works

The economic-based Grid system must contain tools for users and resource providers to convey their needs and goals. In general, there are many resource allocation approaches developed for economic-based Grid systems; these approaches apply some parameters (e.g., deadline, budget and resource cost) to make decisions dynamically at runtime. Economic models could be divided into two categories [1, 2]: the non-price-based-model and the price-based-model. In the non-price-based-model, users share the resources together to meet their requirements; however, in the price-based-model, there are trading systems for users to bid for resources and services. The price-based-model includes commodity markets model, posted price model, bargaining model, tendering/contract-net model and auction model. In the auction model, a resource accepts the bids from many users to acquire it; then, the user making the highest bidding wins the auction. There are several resource allocation algorithms, including the deadline and

budget constrained cost-time optimization algorithm [2, 4, 8], Grid classified optimization scheduling algorithm [9], the Analytic Hierarchy Process and History Based Policy algorithm [10], the Time Optimization algorithm with Auction-Based Proportional Share model [6].

In general, the Time Optimization algorithm with Auction-Based Proportional Share model [6] outperforms other proposed ones. However, it doesn't support the flexible bidding time period, and then it couldn't adjust the priority order to bid in real time. In addition to the lack of the flexibility of the bidding time period, this proposed algorithm [6] conservatively decides the initial bidding costs. The user broker uses the resource cost in the worst case scenario as the initial bidding cost to prevent the job's execution time from exceeding the deadline. But, when the difference of the resource performance between the worst case and the average case is too big, it would cause the abnormal situation that the initial bidding cost excessively exceeds the expected bidding cost, and then wastes the budget.

7.3 Proposed algorithm

In this study, we propose a Flexible Auction-based Resource Allocation (FARA) strategy with a proportional share model for Bag-of-Tasks (BoT) applications. This strategy aims to use lower budgets to complete jobs before their deadlines. The user broker retrieves useable resource in the local Virtual Organization (VO) from the Grid Information System (GIS). The operating steps of our proposed algorithm are listed as follows:

1. Resources register the machine name, the resource capability and the price to the GIS.
2. Each user has a broker which is responsible for scheduling; each user submits tasks, deadline and budget to the broker, and the user broker makes the deployment of communication and the lower level actions.
3. The user broker searches information in the GIS to select appropriate resources for the job after receiving the user's demand.
4. The user broker sends tasks to the selected resource and continues to monitor the status for increasing or reducing the budgets in order to complete jobs in each time period, P . The user broker then pays the total cost to the resource provider.
5. The auctioneer makes the resource work. If the Auctioneer receives the new price sent from the broker in time period, P , it will reallocate the proportion of Grid resources to each user in the next time period.

The framework of the Flexible Auction-based Resource Allocation (FARA) strategy with a proportional share model is shown in Figure 7.1.

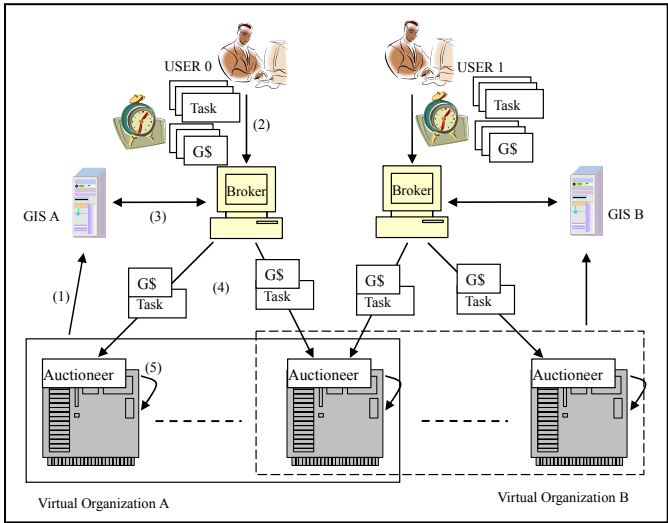


Fig. 7.1. Framework of the proposed FARA algorithm

7.4 Experimental results

The simulation environment is written by JAVA language. The resource characteristics are shown in Table 7.1. There are 5 resources (R1, R2, R3, R4 and R5) and 10 users. The number of jobs for each user is 100. All jobs are generated randomly with the job lengths between 9000~11000 million instructions (MI). The deadline of each user is 1000. In this study, we adopt the ratio of the performance divided by the cost (P/C) as the comparison index for the effectiveness of resource allocation algorithms. That is, the higher the P/C is, the more jobs are completed per unit.

Table 7.1. Resource characteristics

Resource name	R1	R2	R3	R4	R5
MIPS	500	1000	1200	1400	1600
Cost range	1~100	5~500	6~600	7~700	8~800

We assume that user i updates the bid to resume participating the auction every 100 seconds in the time optimization scheduling [6]; however, our proposed FARA algorithm sets the bidding time period, P_i , according to the proportion of shared resources.

The first time period for bidding is set as *est_time*. As for the initial bidding price step, the time optimization scheduling [6] uses the worst case

for the initial bidding (i.e., it assumes that the user completes all jobs with the worst resource only); comparatively, our proposed FARA algorithm uses the average computing cost for the initial bidding. Table 7.2 and Table 7.3 show the simulation results. The P/C of the FARA algorithm is slightly higher than that of the time optimization algorithm; and, the FARA algorithm guarantees to finish all jobs before the deadline. The FARA algorithm allows users to adjust the bid price flexibly and complete all tasks within the deadline. Although part of the tasks would be assigned more expensive resources, the FARA algorithm still guarantees that all the jobs are completed before the deadline.

Table 7.2. Simulation Results of the Time-optimization Scheduling

User	1	2	3	4	5	6	7	8	9	10
Turnaround time	19962	20044	20101	19966	20062	19998	20012	20022	20036	20033
Finishing rate [%]	20.04	19.93	19.86	20.04	19.92	20.00	19.99	19.97	19.95	19.96
P/C	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50

Table 7.3. Simulation Results of the FARA algorithm

User	1	2	3	4	5	6	7	8	9	10
Turnaround time	3915	3920	3919	3933	3928	3930	3939	3924	3936	3942
Finishing rate [%]	100	100	100	100	100	100	100	100	100	100
P/C	0.94	0.93	0.94	0.94	0.94	0.94	0.94	0.94	0.95	0.95

7.5 Concluding remarks

This study proposes a flexible job allocation strategy for economic-based grids to complete jobs before the deadline with lower budgets. This study improves the time-optimization algorithm in the initial auction, and user brokers could flexibly adjust the bidding time and prices to complete jobs. Since the FARA algorithm allows users to adjust the bid price flexibly and thus complete all tasks within the deadline, the simulation results shows that the P/C of the FARA algorithm is higher than that of the time optimization algorithm. Moreover, the FARA algorithm guarantees to finish all jobs before the deadline. The experimental results show that the proposed strategy could outperform the Time Optimization algorithm with Auction-Based Proportional Share model [6].

This study only focuses on CPU resources; however, other resources (such as memory and network bandwidth) could be considered to make the resource allocation more comprehensive.

Acknowledgement. This work was partially supported by the National Science Council of the Republic of China under the contract numbers: NSC 95-2221-E-142-004- and NSC 95-2622-E-142-001-CC3.

References

1. Buyya R, Stockinger H, Giddy J, Abramson D (2001) Economic models for management of resources in peer-to-peer and Grid computing. In: SPIE International Conference on Commercial Applications for High-Performance Computing, Denver, CO
2. Buyya R, Abramson D, Venugopal S (2005) The Grid economy. *Proceedings of the IEEE* 93(3):698-714
3. Chun B, Culler D (2000) Market-based proportional resource sharing for clusters. Technical Report CSD-1092, University of California, Berkeley, CA
4. Foster I, Kesselman C (1999) *The Grid: Blueprint for a future computing infrastructure*. Morgan Kaufmann, San Francisco, CA
5. Pourebrahimi B, Bertels K, Kandru GM, Vassiliadis S (2006) Market-Based Resource Allocation in Grids. In: *Second IEEE International Conference on e-Science and Grid Computing (e-Science'06)*, p 80
6. Sulistio A, Buyya R (2005) A time optimization algorithm for scheduling bag-of-task applications in auction-based proportional share systems. In: *Proceedings of the 17th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, Rio de Janeiro, Brazil, pp 235–242
7. William V (1961) Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance* 16(1):8-37
8. Rajkumar Buyya, Manzur Murshed (2002) A deadline and budget constrained cost-time optimisation algorithm for scheduling task farming applications on global grids. Technical Report, Monash University
9. Li Kenli, Tang Xiaoyong, Zhaohuan (2005) Grid classified optimization scheduling algorithm under the limitation of cost and time. In: *Proceedings of the second International Conference on Embedded Software and Systems*
10. Sai Rahul Reddy P (2006) *Market Economy Based Resource Allocation in Grids*. Master Thesis, Indian Institute of Technology, Kharagpur, India

Part 13

Advances in computer science and applications

In this part we selected some advanced topics in the computer science and its applications. The papers include contributions from a wide range of disciplines - numerical analysis, discrete mathematics and system theory, natural sciences, engineering, information science, electronics, and naturally, computer science itself. The results are addressed not only the specialist in a particular field but also the general scientific audience. The contributions include propagation in cylindrical inset dielectric guide structures, damage in nonlinear mesoscopic materials, diagnosis of robotized assembling systems, open architecture systems for MERO walking robots control, mass transfer analysis in the case of the EFG method, information system architecture for data warehousing, dimensional dynamics identification of reconfigurable machine tools, compaction-induced deformation on flexible substrates, robust and accurate visual odometry by stereo, financial auditing in e-business environment, reputed authenticated routing ad-hoc networks protocol and quantum mechanical analysis of nanowire FETs.

Chapter 1

Propagation in cylindrical inset dielectric guide structures

Ioannis O. Vardiambasis

Microwave Communications & Electromagnetic Applications Laboratory,
Division of Telecommunications, Department of Electronics, Technological Educational Institute (T.E.I.) of Crete - Chania Branch, Romanou 3,
Chalepa, 73133 Chania Crete, Greece, ivardia@chania.teicrete.gr

Abstract. The inset dielectric guide (IDG) is just a rectangular groove filled with dielectric, which retains many of the advantages of dielectric guides, without fabrication and loss problems. Quasi-planar IDGs are attractive for applications in microwave and millimeter-wave frequency ranges, as they can be used as low loss and low cost transmission lines, directional couplers, nonreciprocal phase shifters, antennas and other devices. In this paper we address circular IDG-type structures, investigating hybrid wave propagation in single microslot lines loaded with semi-circular dielectric cylinders. The analysis is based on systems of singular integral-integrodifferential equations of the first kind. The proposed algorithms are rapidly convergent, yielding very accurate results for the propagation constants and for the modal equivalent magnetic currents of the principal as well as the higher-order modes.

Keywords. Hybrid modes, Wave propagation, Microslot lines, Semi cylinders, Dielectric load, Inset Dielectric Guide, Singular equations

1.1 Introduction

Grooves filled with dielectric (Inset Dielectric Guides, IDGs) have been proposed as low loss transmission lines, as directional couplers and band-pass filters, as well as basis for planar antenna arrays [1, 2, 3, 4, 5, 6].

Although rectangular IDGs have been extensively studied in the past using semi-analytical methods (like the transverse resonance diffraction method [1, 2, 3, 4, 5] and the effective dielectric constant method [6]), circular ones have not yet received appropriate attention [7] due to the complexity of the pertinent boundary value problems (mixed circular-plane boundaries) [8, 9]. In this paper we address the latter configurations, combining cylindrical IDG-type structures with microslot lines, which are widely used in modern microwave integrated circuitry. Hybrid propagating modes' characteristics are investigated in single microslots on a ground plane loaded with open or shielded dielectric semi-circular cylinders, as shown in Figure 1.1.

The analysis, briefly discussed in Section 1.2, is based on exact systems of coupled singular integral-integrodifferential equations of the first kind, which are efficiently discretized using the methods explored in [8, 9]. More details on the formulation and the method of solution may be found in [9] dealing with 3D scattering by these same structures. The proposed algorithms yield very accurate results for the propagation constant as well as for the modal electric currents of all modes, as will be seen in Section 1.3 where such typical results are presented for several cases.

1.2 Formulation

Figure 1.1 shows a microslot line of width $2w$, covered by two semi-circular dielectric cylinders of radii α_1 and α_2 (regions 1 and 2).

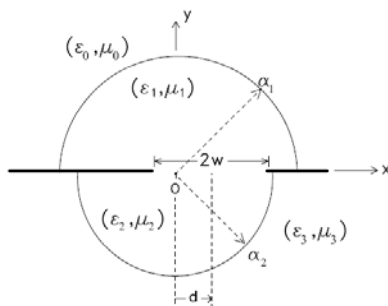


Fig. 1.1. Geometry of a slotline loaded with two semi-cylinders

Each of the regions 0 ($r > \alpha_1$, $\varphi > 0$) and 3 ($r > \alpha_2$, $\varphi < 0$) may be either air or perfect electric conductor (PEC). Each of the regions is characterized by its dielectric permittivity ϵ_s and its magnetic permeability μ_s ($s=0,1,2,3$). For simplicity in notation the semi-cylinders are taken to be coaxial with their common axis displaced from the slot axis by d .

Assuming propagation in the $-\hat{z}$ direction, the slotline's equivalent surface magnetic current densities $M_z(x) \cdot e^{j(\omega t + \beta z)}$ and $M_x(x) \cdot e^{j(\omega t + \beta z)}$ satisfy the following homogeneous system of coupled singular integral-integro-differential equations [9]:

$$\sum_{s=1}^2 \left\{ -\frac{k_{cs}^2}{2\omega\mu_s} \int_C M_z(x) \cdot H_0^{(2)}(k_{cs} |x - x'|) dx - \right. \quad (1.1)$$

$$-\frac{j\beta}{2\omega\mu_s} \frac{d}{dx'} \int_C M_x(x) \cdot H_0^{(2)}(k_{cs} |x - x'|) dx +$$

$$+ \sum_{n=0}^{\infty} \epsilon_n J_n(k_{cs} x') \left[2B_n^M \int_C M_z(x) \cdot J_n(k_{cs} x) dx - \right.$$

$$\left. -\frac{1}{k_{cs}} \int_C M_x(x) \left(j\beta B_n^M \cdot h_n^-(k_{cs} x) + \omega\epsilon_s A_n^M \cdot h_n^+(k_{cs} x) \right) dx \right] \Bigg\} = 0 \quad (x' \in C)$$

$$\sum_{s=1}^2 \left\{ -\frac{j\beta}{2\omega\mu_s} \frac{d}{dx'} \int_C M_z(x) \cdot H_0^{(2)}(k_{cs} |x - x'|) dx - \right. \quad (1.2)$$

$$-\frac{1}{2\omega\mu_s} \left[\frac{d^2}{dx'^2} + k_s^2 \right] \int_C M_x(x) \cdot H_0^{(2)}(k_{cs} |x - x'|) dx +$$

$$+ \frac{1}{k_{cs}} \sum_{n=0}^{\infty} \epsilon_n \left[\left(j\beta B_n^M h_n^-(k_{cs} x') - \omega\epsilon_s B_n^J h_n^+(k_{cs} x') \right) \cdot \int_C M_z(x) \cdot J_n(k_{cs} x) dx + \right.$$

$$+ \frac{1}{2k_{cs}} \int_C M_x(x) \left[\beta h_n^-(k_{cs} x') \left(\beta B_n^M h_n^-(k_{cs} x) - j\omega\epsilon_s A_n^M h_n^+(k_{cs} x) \right) + \right.$$

$$\left. \left. + \omega\epsilon_s h_n^+(k_{cs} x') \left(\omega\epsilon_s A_n^J h_n^+(k_{cs} x) + j\beta B_n^J h_n^-(k_{cs} x) \right) \right] dx \right] \Bigg\} = 0 \quad (x' \in C)$$

where β is the unknown propagation constant, $H_0^{(2)}(\cdot)$ is the Hankel function of second kind and of zero order, $J_n(\cdot)$ is the Bessel function of order n , C denotes the slot interval $d - w \leq x' \leq d + w$, $k_{cs}^2 = k_s^2 - \beta^2$, $k_s^2 = \omega^2 \epsilon_s \mu_s$, $\epsilon_n = 2 - \delta_{n0}$ is the Neumann's factor (δ_{nm} is the Kronecker

delta), $h_n^\pm(\cdot) = J_{n-1}(\cdot) \pm J_{n+1}(\cdot)$, whereas $A_n^M \equiv A_n^M(s, s)$, $B_n^M \equiv B_n^M(s, s)$, $A_n^J \equiv A_n^J(s, s)$, $B_n^J \equiv B_n^J(s, s)$ are specified in the Appendix of [9]. Note that by $s=1, 2$ we denote the regions (semicylindrical or half plane) from both sides of the microslotline.

The solution of the system (1.1)-(1.2) is carried out on a moment method basis using expansions of the magnetic current distributions in terms of first and second kind Chebyshev polynomials. Following singular integral equation techniques in the way outlined in [9], we arrive at the infinite system of linear algebraic equations:

$$\sum_{N=0}^{\infty} [a_N R_{MN}^{zz} + b_N R_{MN}^{zx}] = 0, \sum_{N=0}^{\infty} [a_N R_{MN}^{xz} + b_N R_{MN}^{xx}] = 0; M = 0, 1, \dots, \infty \quad (1.3)$$

The matrix elements R_{MN}^{uv} ($u, v \equiv z, x$) assume rapidly converging analytical expressions given in [9]. Non-zero solutions of (1.3), for the expansion constants a_N and b_N of the slotline's equivalent surface magnetic currents, require vanishing of its determinant. This yields to the dispersion equation of the structure. The propagation constants and equivalent current distributions of the principal as well as of the higher order modes can be efficiently obtained with very high accuracy for any combination of the physical and geometrical parameters of the configuration.

1.3 Numerical results

In single-slot configurations loaded with open dielectric semi-cylinders there are always two (for single load) or three (for double load) principal modes, which are never cutoff in analogy with [8]. These include: (a) the slot mode (S-mode), and (b) the HE_{11} modes from the regions 1 and 2.

In the special case ($\epsilon_0 \rightarrow \infty$, $\epsilon_3 \rightarrow \infty$, $\alpha_1 = \alpha_2$) all principal as well as all higher order modes coincide with the corresponding modes of the circularly shielded microslot lines studied in [10]. This coincidence of results may be used as a validity check for the developed here numerical codes.

In Figure 1.2, where some typical results are presented in case of a symmetrical slot ($d=0$), there are mode transformation phenomena that take place between the two principal modes as well as between the 4th, 5th and 6th modes. Such phenomena do not occur in case $d \neq 0$, as illustrated in Figure 1.3, where the S-mode has the highest ϵ_{eff} over the whole frequency range.

1.4 Conclusions

Generalized cylindrical IDG structures, or microslot lines loaded by one or two semicircular dielectric cylinders, could be useful in microwave applications. The analysis of this structure was based on exact singular integral equation techniques. The algorithms that have been developed converge rapidly and can be used to obtain the propagation constant and the modal currents of any mode with high accuracy.

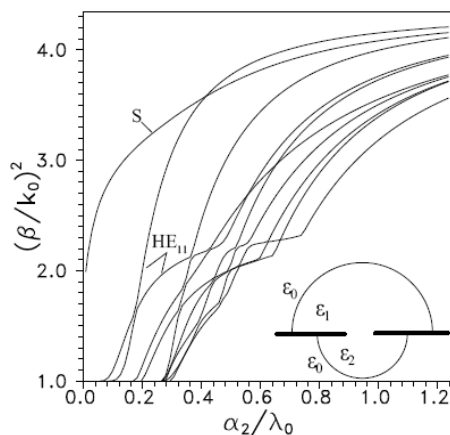


Fig. 1.2. Effective dielectric constant $\epsilon_{\text{eff}} = (\beta/k_0)^2$ versus normalized frequency for the first 10 propagating modes of the structure of Figure 1.1 with $2w = 0.5\alpha_2$, $d = 0$, $\epsilon_1 = 2.32\epsilon_0$, $\epsilon_2 = 4.34\epsilon_0$, $\epsilon_3 = \epsilon_0$, $\mu_1 = \mu_0$, $\alpha_1 = 2\alpha_2$

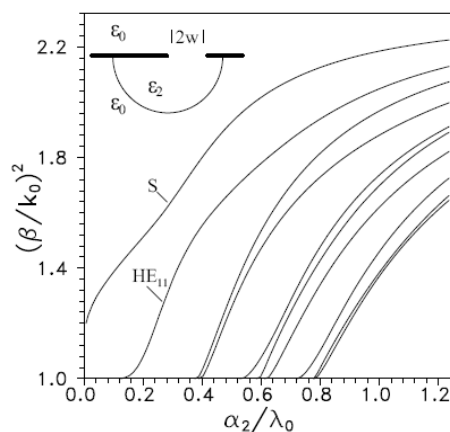


Fig. 1.3. ϵ_{eff} vs α_2/λ_0 for the first 10 propagating modes of the structure of Figure 1.1 with $2w = 0.5\alpha_2$, $d = 0.5\alpha_2$, $\epsilon_1 = \epsilon_3 = \epsilon_0$, $\epsilon_2 = 2.32\epsilon_0$, $\mu_1 = \mu_0$

Acknowledgments. This work is co-funded by the European Social Fund (EKT) with 75% and National Resources (Greek Ministry of National Education and Religious Affairs) with 25% under the ΕΠΕΑΕΚ project “Archimedes – Support of Research Groups in TEI of Crete – 2.2.-7 – Smart antenna study and design using techniques of computational electromagnetics and pilot development and operation of a digital audio broadcasting station at Chania (SMART-DAB)”.

Also, the author would like to thank Prof. John L. Tsalamengas and Ass. Prof. George Fikioris, of the National Technical University of Athens, for their helpful discussions and insightful comments.

References

1. Pennock SR, Izzat N, Rozzi T (1992) Very wideband operation of twin-layer inset dielectric guide. *IEEE Trans Microwave Theory Techn* 40:1910-1917
2. Rozzi T, Hedges SJ (1987) Rigorous analysis and network modeling of the inset dielectric guide. *IEEE Trans Microwave Theory Techn* 35:823-833
3. Rozzi T, Ma L (1988) Mode completeness, normalization, and Green's function of the inset dielectric guide. *IEEE Trans Microwave Theory Techn* 36:542-551
4. Rozzi T, Ma L, Leo R, Morini A (1990) Equivalent network of transverse dipoles on inset dielectric guide (IDG): Application to linear arrays. *IEEE Trans Antennas Propagat* 38:380-385
5. Sewell PD, Rozzi T (1993) The continuous spectrum of the inset dielectric guide and its application to waveguide transitions. *IEEE Trans Microwave Theory Techn* 41:282-289
6. Zhou W, Itoh T (1982) Analysis of trapped image guides using effective dielectric constant and surface impedances. *IEEE Trans Microwave Theory Techn* 30:2163-2166
7. Vardiambasis IO, Tsalamengas JL, Fikioris JG (1998) Hybrid wave propagation in microslot lines loaded with dielectric semi-cylinders. In: 1998 URSI International Symposium on Electromagnetic Theory Proceedings 1:163-165
8. Vardiambasis IO, Tsalamengas JL, Fikioris JG (1997) Hybrid wave propagation in generalised Goubau-type striplines. *IEE Proc Microwaves Antennas Propag* 144:167-171
9. Vardiambasis IO, Tsalamengas JL, Fikioris JG (1998) Plane wave scattering by slots on a ground plane loaded with semicircular dielectric cylinders in case of oblique incidence and arbitrary polarization. *IEEE Trans Antennas Propagat* 46:1571-1579
10. Vardiambasis IO, Tsalamengas JL, Fikioris JG (1995) Hybrid wave propagation in circularly shielded microslot lines. *IEEE Trans Microwave Theory Techn* 43:1960-1966

Chapter 2

On the damage in nonlinear mesoscopic materials

Veturia Chiroiu, Dan Dumitriu, Ana Maria Mitu, Daniel Baldovin, Cornel Secara

Institute of Solid Mechanics, Ctin Mille 15, Bucharest 010141

Abstract. This paper is devoted to the analysis of the damage in nonlinear mesoscopic materials, which are aggregates of grains which act as rigid vibrating units, while the contacts between them - the bond system - constitute a set of interfaces that control the behaviour of the material. The interfaces are mesoscopic, with a typical size of 1 micrometer. A constitutive micropolar model for material interfaces is presented. The model is based on damage coupled to plastic or viscoplastic slip, stick and dilatation (separation), and it is able to describe the successive degradation and failure of an interface.

Keywords. Mesoscopic material, micropolar theory, damage, interface, viscoplasticity

2.1 Introduction

Damaged materials form a class of unusual elastic materials that show extreme nonlinearity, hysteresis and discrete memory. This class includes pearlitic steel, fiber-reinforced metal matrix composites, cement, concrete, ceramics, rocks, sand, soil etc. Materials whose elastic properties are determined by such a bond system are called nonlinear mesoscopic materials.

The bond systems consist of a fabric of defects (cracks, voids) that participates in the elastic response of the material. The grains have a random

position and the intergranular interfaces contain cavities, microvoids, defects. The mesoscopic materials cannot be described by the traditional theory of elasticity [1, 2]. The degradation and deterioration on the microstructural level may be viewed as the successive evolution with load and time of microdefects in the shape of microvoids and microcracks, which are part of the microstructure heterogeneity. For example, residual microstresses after the manufacturing processes, effects form the large precipitates present in grey-cast iron are initiators of microcracks [3]. This paper is devoted to the analysis of the damage in nonlinear mesoscopic materials, which are aggregates of grains which act as rigid vibrating units, while the contacts between them - the bond system - constitute a set of interfaces that control the behaviour of the material. The interfaces are mesoscopic, with a typical size of $1\text{ }\mu\text{m}$ [4-6]. At the mesoscopic scale, the micropolar continuum mechanics possesses a great potential to describe the long-range interactions among the particles in materials [7-11].

2.2 Basic micropolar equations

Throughout this paper a rectangular coordinate system (x_1, x_2, x_3) is employed and repeated indices denote summation over the range (1,2,3). Balance of momentum equation is written as $t_{ji,i} + \rho(f_i - \dot{v}_i) = 0$, where t_{ij} is the stress tensor, f_i the body force vector, v_i the velocity vector, ρ the density, a superposed dot indicates material differentiation, i.e. $\dot{v}_i = v_{i,t} + v_k v_{i,k}$ and a comma represents partial differentiation with respect to the variable. The balance of moment of momentum equation is $m_{ji,j} + e_{ijk} t_{jk} + \rho(l_i + j\dot{\phi}_i) = 0$, where m_{ij} is the couple stress tensor, l_i is the body couple vector, v_i is the microgyration vector which is also the material derivative of the microrotation vector ϕ_i , j is the micro-inertia, and e_{ijk} is the permutation tensor. The constitutive law can be written as

$$t_{ij} = \lambda \varepsilon_{kk} \delta_{ij} + (\mu + \kappa) \varepsilon_{ij} + \mu \varepsilon_{ji},$$

$$m_{ij} = \alpha x_{kk} \delta_{ij} + \beta x_{ji} + \gamma x_{ij}, \quad x_{ij} = \phi_{j,i},$$

where δ_{ij} is the Kronecker delta and the microstrain tensor ε_{ij} links the displacement vector u_i to the microrotation vector through the relations $\varepsilon_{ij} = u_{j,i} + e_{jik} \phi_k$. The kinematic characteristics of the strain are given by

ε_{ij} and x_{ij} . Six micropolar elastic moduli are introduced: λ and μ are the Lamé constants from classical elasticity, and κ , α , β and γ are additional constants in micropolar theory [7].

The above formulation is complete since there are 33 equations in the 33 variables t_{ij} , m_{ij} , ε_{ij} , ϕ_i and u_i . As boundary conditions, we can prescribe the surface tractions and couples: $t_{(n)i} = t_{ji}n_j$, $m_{(n)i} = m_{ji}n_j$ on S , where $t_{(n)i}$ and $m_{(n)i}$ are the surface stress and couple vectors, respectively, the subscript (n) referring to the bounding surface S whose exterior unit normal vector is n_j . It is also possible to prescribe the displacement u_i and the microrotation ϕ_i on S : $\phi_i = \phi_i^0$, $u_i = u_i^0$ on S , or to construct mixed boundary conditions in displacements and microrotations over a portion of the surface S_1 and tractions and couples over the remainder surface S_2 , $S = S_1 \cup S_2$, $S_1 \cap S_2 = \emptyset$.

2.3 Proposed model for the interface

Let us consider an interface in the shape of a band of width δ , in a local coordinate system, along which the micro constituents A and B are not perfectly bonded, because of the slip and dilatation which may occur along the interfaces (Figure 2.1). Let q_i and s_i be the traction vector and respectively the couple vector acting on the body B defined by the normal $n_i = (1, 0, 0)$, $q_i = t_{ij}n_j = t_{i1}$, $s_i = m_{ij}n_j = m_{i1}$, $i = 1, 2, 3$, where q_i and s_i are continuous along the interface: $[q_i] = 0$, $[s_i] = 0$, $i = 1, 2, 3$ (the bracket indicates a jump across the interface $[q_i] = q_i^A - q_i^B$).

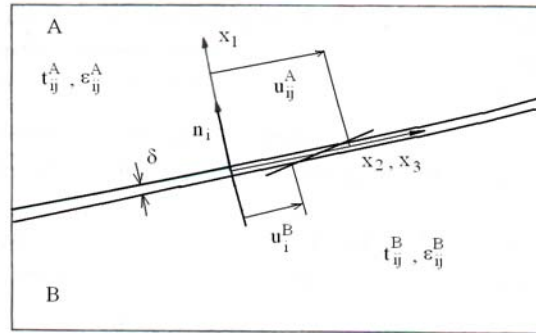


Fig. 2.1. Interface between the bodies A and B

It results that the components t_{11}, t_{12}, t_{13} and m_{11}, m_{12}, m_{13} are always continuous and these unique values may be associated with the interface. For the remaining components t_{22}, t_{23}, t_{33} and m_{22}, m_{23}, m_{33} it is not possible to associate unique values with the interface. So, in a manner similar to [3], the independent uniquely defined state variables at the interface are

$$t_{ij} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{12} & a & a \\ t_{13} & a & a \end{bmatrix}, \quad m_{ij} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{12} & b & b \\ m_{13} & b & b \end{bmatrix}, \quad \varepsilon_{ij} = \begin{bmatrix} c & c & c \\ c & \varepsilon_{22} & \varepsilon_{23} \\ c & \varepsilon_{23} & \varepsilon_{33} \end{bmatrix},$$

where a, b, c are non-unique variables. For variables a, b , thermodynamically associated stresses and couple-stresses that are energy conjugates to $\varepsilon_{22}, \varepsilon_{23}, \varepsilon_{33}$, are introduced. For variable c , thermodynamic strains and microstrains that are energy conjugates to $t_{11}, t_{12}, t_{13}, m_{11}, m_{12}, m_{13}$, are introduced. It follows

$$\tilde{t}_{ij} = \begin{bmatrix} \tilde{a} & \tilde{a} & \tilde{a} \\ \tilde{a} & \tilde{t}_{22} & \tilde{t}_{23} \\ \tilde{a} & \tilde{t}_{23} & \tilde{t}_{33} \end{bmatrix}, \quad \tilde{m}_{ij} = \begin{bmatrix} \tilde{b} & \tilde{b} & \tilde{b} \\ \tilde{b} & \tilde{m}_{22} & \tilde{m}_{23} \\ \tilde{b} & \tilde{m}_{23} & \tilde{m}_{33} \end{bmatrix}, \quad \tilde{\varepsilon}_{ij} = \begin{bmatrix} \tilde{\varepsilon}_{11} & \tilde{\varepsilon}_{12} & \tilde{\varepsilon}_{13} \\ \tilde{\varepsilon}_{12} & \tilde{c} & \tilde{c} \\ \tilde{\varepsilon}_{13} & \tilde{c} & \tilde{c} \end{bmatrix}.$$

In order to define these thermodynamic variables, we consider a relative motion in the interface to couple kinetically the motion with damage. We introduce the jump of displacements u_i and the microrotation ϕ_i across the interface, defined as $u_i^0 = u_i^A - u_i^B$, $\phi_i^0 = \phi_i^A - \phi_i^B$ and $\varepsilon_{1j} = u_{j,1} + e_{j1k} \phi_k$, $x_{1j} = \phi_{j,1}$.

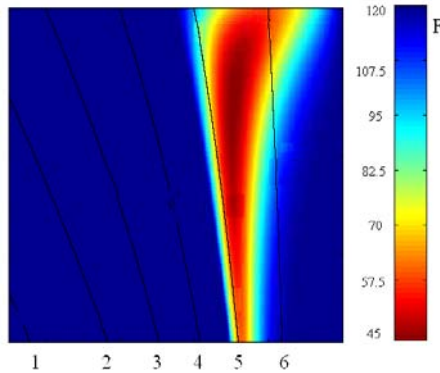


Fig. 2.2. The field of forces showing the successive localization and shear band development

2.4 Degradation and failure of the interface

To analyze the successive degradation and failure of an interface as shown in Figures 2.1 and 2.2, we consider that this interface is subjected to monotonic loading ([12-16]). The states 1-6 in Figure 2.2 refer to the corresponding field of forces which show the successive localization and shear band development in the interface.

2.5 Conclusions

The propagation of the interface damage into materials A and B is schematically illustrated in Figure 2.3. The damage of the interface affects also the state of integrity of the adjacent bodies. The lines represent the directions in which the cracks initiated into the interface move outside the interface into bodies A and B.

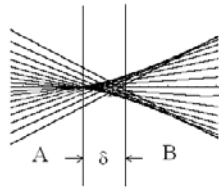


Fig. 2.3. The propagation of cracks into the bodies A and B



Fig. 2.4. Intragranular minicraters of damage

The failure of the interface generates also an avalanche of plastic minicraters into the material as shown in Figure 2.4. These minicraters are observed macroscopically along the grain boundaries at ferrous alloys such as steel with a pearlitic ministructure. They can be attributed to the failure in the sense of nucleation, growth and coalescence [3], [17], [18].

Acknowledgement. The authors acknowledge the financial support of the CEEEX postdoctoral grant 1531/2006. Best thanks are presented also to Professor *Pier Paolo Delsanto* and to Dr. *Marco Scalerandi* (Dipartimento di Fisica, Politecnico di Torino) for the important suggestions and discussions in the field of this topic.

References

1. Guyer RA, McCall KR, Boitnott GN (1995) Hysteresis, discrete memory, and nonlinear wave propagation in rock. *Phys.Rev.Lett.* 74:3491–3494
2. Guyer RA, Johnson PA (1999), Nonlinear mesoscopic elasticity: Evidence for a new class of materials. *Physics Today*
3. Cannmo P (1997) A damage based interface model: Application to the degradation and failure of a polycrystalline microstructure. Chalmers Univ. of Techn., Div. of Solid Mechanics, Göteborg, Sweden, pp 1–64
4. Kachanov LM (1958) Time of the rupture process under creep conditions. *Izv. Akad., SSSR, Otd. Tech. Nauk.* 8:26–31
5. Lemaître J (1985) A continuous damage mechanics model for ductile fracture. *ASME, J. Eng. Materials Technology* 107:83–89
6. Lemaître J (1990) Micromechanics of crack initiation. *Int. J. Fracture* 42:87–99
7. Eringen AC (1966) Linear theory of micropolar elasticity. *J. Math. & Mech.* 15:909–924
8. Eringen AC (1968) Theory of micropolar elasticity. In: Liebowitz R (ed) *Fracture*. Academic Press, 2:621–729
9. Mindlin RD (1964) Microstructure in linear elasticity. *Arch. Rat. Mech. Anal.* 16:51–78
10. Kröner E (1963) On the physical reality of torque stresses in continuum mechanics. *Int. J. Engng. Sci.* 1:261–278
11. Berglund K (1982) Structural models of micropolar media. In: *Mechanics of Micropolar Media*, World Scientific, Singapore
12. Gauthier RD (1962) Experimental investigations on micropolar media. In: Brulin O, Hsieh RKT (eds) *Mechanics of Micropolar Media*. World Scientific, pp 395–463
13. Lakes RS (1986) Experimental microelasticity of two porous solids. *Int. J. Solids, Structures* 22:55–63
14. Badea T, Chiroiu C, Soare M, Chiroiu V (2000) The inverse problem of micro-defects identification. *Rev. Roum. Sci. Techn.* 45(1):31–39
15. Badea T, Nicolescu M (2004) A Preisach model of hysteretic behavior of nonlinear elastic materials. In: Chiroiu V, Sireteanu T (eds) *Topics in Applied Mechanics*. Ed. Academiei Bucharest, 1:1–25
16. Chiroiu V (2004) Identification and inverse problems related to material properties and behaviour. In: Chiroiu V, Sireteanu T (eds) *Topics in Applied Mechanics*. Ed. Academiei Bucharest, 1:83–126
17. Teodorescu PP, Munteanu L, Chiroiu V (2005) On the wave propagation in chiral media. *New Trends in Continuum Mechanics*, Ed. Theta Foundation, Bucharest, pp 303–310
18. Teodorescu PP, Badea T, Munteanu L, Onişoru J (2005) On the wave propagation in composite materials with a negative stiffness phase. *New Trends in Continuum Mechanics*, Ed. Theta Foundation, Bucharest, pp 295–302

Chapter 3

Diagnosis of robotized assembling systems

C. Ciufudean,¹ R. Serban,² C. Filote,¹ B. Satco¹

¹University “Stefan cel Mare”, Faculty of Electrical Engineering and Computer Science, 9 University str., Suceava, Romania, calin@eed.usv.ro

²Instituute d’Ingenierie Informatique de Limoges et de Rodez,
43 rue Sainte-Anne, Limoges, France, rares.serban@3il.fr

Abstract. In this paper we propose a Petri net within a diagnosis system for robot construction design. We assimilate the robot construction process with an assembly process that composes parts and/or subassemblies into a unique product. We assume that the assembly supervisor (AS) system is distributed, and it solves several local AS’s attached to the nodes of the Petri net model of the assembly process. The research issues that we address in this work include the modelling of assembly process, the determination of cost-effective assembly plans for efficient building, and the real time adoption of a plan to a given product to be assembled. This work extends the known assembly Petri nets to a powerful framework enabling to derive the diagnosis of assembly processes whose path may vary, and the objective function is maximized. The presented approach can be used to evaluate transient and steady state performances of alternative design based on a construction example.

Keywords. Petri nets, diagnosis, assembly process, objective function, state estimation function

3.1 Introduction

In this paper we focus on the diagnosis of asynchronous systems. Typical examples are robot-assembled systems, such as shown in Figure 3.1, where the supervisor system is distributed, and it involves several local supervisors, attached to some nodes of the construction network. Each local supervisor has only a partial view of the overall system. The different local supervisors have their own local time, but they are not synchronized. Alarms are reported to the global supervisor, and this supervisor performs the diagnosis. We notice that events may be correctly ordered by local supervisors, but communicating observations via network causes a loss of synchronization, and results in a non-deterministic supervisor [1-6]. Alternative process plans can be exploited in real time to react to machine failures, in order to avoid having bottleneck machines, and to enable adaptive production planning of failure-prone construction systems. The work proposes an adaptive process-planning scheme that can manage process changes and adapt the process to specific assembly conditions. In order to solve this problem, the paper proposes a methodology for the design and implementation of an adaptive assembly planner based on assembly Petri nets (APN).

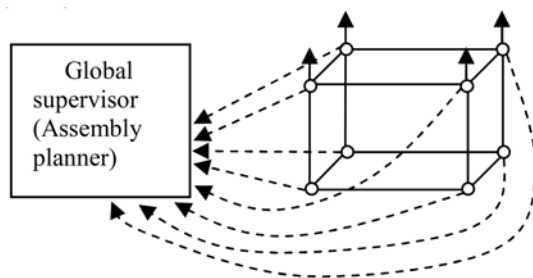


Fig. 3.1. Supervision of a construction system

The advantages of using APN's include the following [7,8]: allowing the dynamic behaviour to be visualized; representing both the assembly process and system resources in a single presentation for diagnosis and easy control implementation; allowing a linear programming formulation to find optimal assembly plans. The proposed planner integrates, as shown in Figure 3.1, an assembly system, and this principia scheme is represented in Figure 3.2. The input to the system consists of raw materials; the output is the finite product (construction) and what remain to be dumped, secondary raw materials, s.a. During assembly execution, observations made by local supervisors are transferred directly to the global supervisor (assembly

planner - see Figure 3.1). They are used to update the predicted values of each component and respective assembly costs. Hence, the assembly system adapts the predictive process plan to the new data and generates an adapted plan that may lead to a new termination goal. The objectives of this paper are to present a method for developing an adaptive planner and to illustrate assembly process planning via a specific design with execution success rate and respective costs. Section 3.2 describes the proposed robot assembled Petri net model and planning algorithm, respectively the assembly supervisor Petri nets (ASPN). Section 3.3 deals with an important problem concerning the Petri nets modelling process, respectively the size of the Petri net model. This paper presents an approach that decomposes the ASPN into several basic configurations (e.g., subnets), in order to determine the state estimation function (Se-function) that describes the Petri net state-space size. This approach allows the designer to choose a proper ASPN model for a complex robot assembled system. Section 3.4 presents a design and implementation methodology for a robot assembled system.

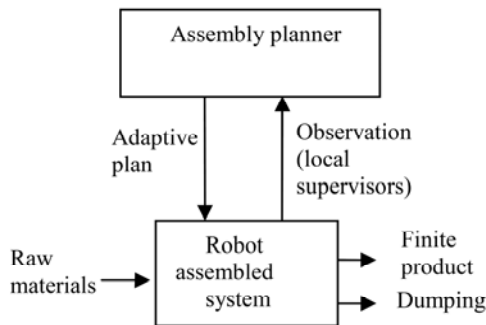


Fig. 3.2. Adaptive robot assembled system

2.2 Modelling the robot assembled process: Assembly supervisor Petri nets

This paper extends the known assembled Petri nets (APN) into assembly supervisor Petri nets (ASPN). Thus ASPN's can accurately describe the robot construction topology, mating relations and precedence relations. In an ASPN a transition (assembly process) and a place (a product, or its subassembly) are associated with utility information (cost/benefit). Each transition is also associated with pre-firing and post-firing values. The pre-firing value is a decisional value which indicates the priority level for a transition to fire [8], respectively its associated assembly operation to perform. The

post-firing value represents a probability that indicates the success rate of its assembly operation, which is updated based on the observations received from the local supervisors. The ASPN can estimate the assembly performance, e.g. net profit, and also decides the best actions among various corrective actions, in order to maximize the profit. An assembly supervisor Petri net is defined as: $ASPN = (P, T, W, Mo, f_1, f_2, v_d, v_p)$, where P and T are finite sets of places and transitions, respectively; $W \subseteq (P \times T) \cup (T \times P)$ is a set of directed arcs; $Mo: P \rightarrow N$ is the initial marking, where N is the set of nonnegative numbers. The set of input (output) transitions of a place $p \in P$ is denoted by ${}^0p$ (p^0). The set of input (output) places of a transition $t \in T$ is denoted by ${}^0t$ (t^0) [5, 6]. We also have:

- $f_1: P \rightarrow R^+$ is the resource utility function assigned to a place, where R is the set of nonnegative real numbers;
- $f_2: T \rightarrow R^+$ is the cost function assigned to a transition, where R^+ is the set of nonnegative real numbers;
- $v_d: T \rightarrow N$ is a decisional value assigned to a transition. This value is assigned according to a planning algorithm. Value v_d is used to decide the firing priority of the transitions;
- $v_p: T \rightarrow [0,1]$ is a probability value associated with a transition, that is updated according to the sensing result of the corresponding assembly operation. The value $v_p(t)$ represents the success rate of an assembly operation. The value $1 - v_p(t)$ represents the failure rate.

We notice that in an ASPN model, a place with multiple output transitions represents a subassembly with multiple ways to be assembled. These different assembly choices should determine a common set of assembled parts. Multiple output transitions from a place form a Logic-OR relation and multiple output places from a transition form a Logic-AND relation. Both place and transition utility functions are used to generate an optimal assembly plan. The decision and probability value of transitions are used to execute, respectively to adapt the construction plan. ASPN defined in such a way belongs to the class of free-choice Petri nets [9]. That means that the global ASPN model has the structure given in Figure 3.3, where P_{si} , $i = 1, \dots, n$ are the partial subassemblies of the construction, and P_f represents the final product. Obviously, a construction planning schedule is to determine the best order of assembly operations, i.e. transition firings. In order to reject the uncertainty of the assembly operations, different-level priorities are assigned to different assembly alternatives for all the subassemblies [10, 11].

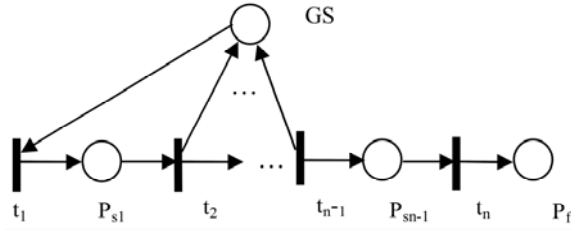


Fig. 3.3. Global ASPN structure

In an ASPN, a place with multiple transitions implies assembly methods, each of them having its own value (v_w). Introducing the v_d (decisional value) to each transition enables an easy determination of assembly order. For example let place p having k output transitions: $t_1, t_2, \dots, t_k$ and their values v_{wi} , $i = 1, \dots, k$. The priority levels to transitions t_i are assigned accordingly to the following relation:

$$v_d(t_j) = i \quad (3.1)$$

Where v_{wi} is the i^{th} smallest among v_{wi} , $i = 1, \dots, k$. When an assembly operation fails (e.g. the ASPN diagnosis is revealing a bottleneck or a too expensive way of firing transitions), the assembly planner selects a transition with the next largest v_d value, and so on. The v_p values assigned to transitions are designed to adapt the ASPN for the maximum expected assembled value. Initially, all v_p are set accordingly to assembly planner (designer) experience. During execution, for each assembly operation, the number of successes are recorded and different v_{pi} , $i = 1, \dots, k$ are re-adjusted with an exponential rate $e^{-N/N_s t_i}$ where N is the number of transitions fired for a subassembly, and N_s is the number of successes. We notice that v_d is assigned based on both v_p and v_w for each transition.

3.3 SE – Functions for Basic Configurations of ASPN

According to the assumptions made in the previous section, we define a generally firing policy for the ASPN:

$$a_{it} = \min v_{pj}, \forall i, j \in \{T\} \quad (3.2)$$

$$T_j/m_{it} = 0$$

where a_{it} is the success rate of firing transition i at time t ; m_{it} is the marking of place p_i at time t ; T_j is the transition which follows the place p_i with marking m_{it} ; v_{pj} is the value v_p assigned to transition T_j .

We determine the size of the state-space for the underlying system, irrespective of his dimension, by decomposing it into several basic configurations, e.g. subnets (SN) that can form complex (large) ASPN, while combined interconnections for mechanisms of execution are presented here. ASPN can model the execution of sequential, parallel and choice operations. This approach allows the designer to choose a properly ASPN model for a complex robot assembled system. Figure 3.4 illustrates two subnets in series: SN₁, respectively SN₂.

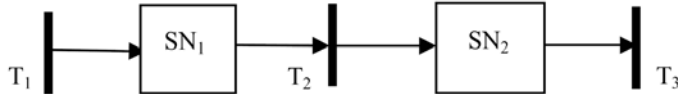


Fig. 3.4. Series basic interconnection of ASPN

The possible distribution of r tokens among the places of the subnets SN₁, and SN₂ is due to the marking repartition among the respective places, separated by transitions T_i , $i=1,2,3$, with a_{it} success rate of firing. The interconnection Se-function is given by the next relation [12, 15]:

$$Se_{series} = \sum_{i=0}^r Se_{a_{1t}}(i) Se_{a_{2t}}(r-i) \quad (3.3)$$

where $Se_{a_{nt}}(i)$ is the k^{th} partition of r , is an n -vector of non-negative integers that sum to r , $i=1,2$, at time t . For n subnets, the Se-function is:

$$Se_{series} = \sum_{i=0}^r Se_{a_{1t}}(i) Se_{a_{2t}}(i-1) \dots Se_{a_{i-1t}}(r-i+1) \cdot Se_{a_{it}}(r-i) \quad (3.4)$$

Figure 3.5 depicts the parallel execution of the operations. For every token entering through T_1 , there is one in each subnet. The Se-function is given by the next relation [11, 13]:

$$Se_{parallel} = \prod_{i=1}^n Se_{a_{it}}(r) \quad (3.5)$$

The choice among sub-nets is represented in Figure 3.6. We consider three configurations: SN₁, respectively SN₂ and, as a group, the places P_1 and P_2 . Having r tokens among P_1 and P_2 generates $C_r^2 = r+1$ states. For n sub-nets, we have:

$$Se_{choice} = \sum_{i=0}^n Se_{1_{a1t}}(i) Se_{2_{a2t}}(i-1) \dots Se_{a_{i-1t}}(r-i+1) \cdot Se_{a_{it}}(r-i) \cdot (i+1) \quad (3.6)$$

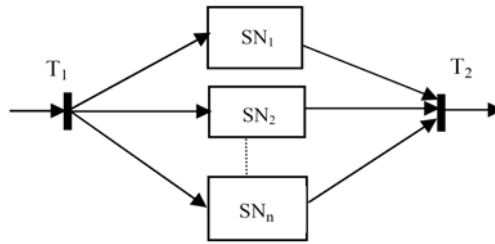


Fig. 3.5. Parallel basic interconnection of ASPN

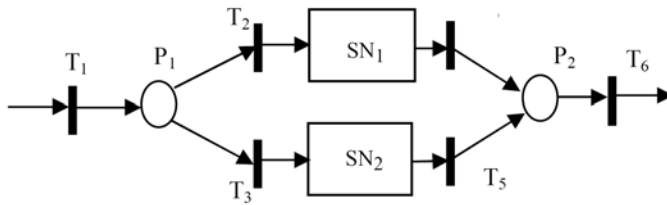


Fig. 3.6. Free choice basic interconnection of ASPN

ASPN of systems containing unreliable components may include models for the failure and repair of these components. Figure 3.7 depicts a common model for these operations. Under normal functioning, T_2 fires instead of T_{fail} , leaving SN_2 out of consideration. The choice between T_2 and T_{fail} allows SN_2 to use all r tokens. Thus, SN_1 and SN_2 are, in series, producing the following Se-function:

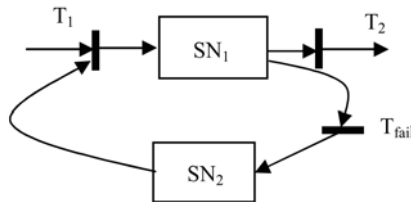


Fig. 3.7. Failure/repair interconnection of ASPN

3.4 Design and Implementation Method

Using the proposed ASPN and algorithms, we have the following design and implementation steps for a construction system: a) Construct an ASPN given the product information and the properly Se-function; b) Associate all the data with places and transitions in the ASPN; c) Run the ASPN based on the construction resources. In order to understand these steps we

implement an assembly (construction) system as a plausible example. We have raw material type A,B,C,D,E (e.g. parts A,B,C,D,E). The possible way to assembly these parts to obtain the final subassembly F is depicted in Figure 3.8. We noted with TD the places that symbolize the dumping materials. In Figure 3.8, for each location we assign the utility function $f_1(p_1 \div p_{11}) = (1,2,3,...,11)$. For example, the final product obviously has the greatest value, and the other values were assigned arbitrarily in this application. The assembly cost in this example is as follows: $f_2(t_1 \div t_9) = (1,2,...,9)$. Each transition has allocated in Figure 3.8 the respective cost. Once we have the cost/benefits values of places and transitions we can find the optimal assembly plan (e.g. the optimal way in ASPN). The general job-shop scheduling problem has been shown to be NP-complete. Therefore, we resort to the heuristic search algorithm to solve this problem. We use a heuristic best-first search procedure known as A* algorithm [10].

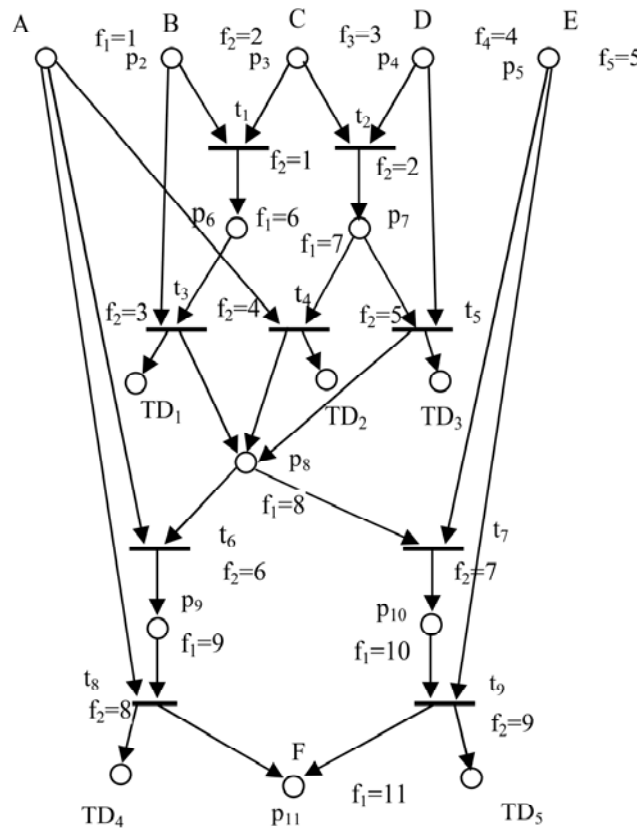


Fig. 3.8. ASPN of a construction subassembly

This algorithm is the following one:

- Step 1. Place initial marking M_0 on the list VALID
- Step 2. If VALID is empty terminate with failure.
- Step 3. Choose a marking M from the list VALID with maximal cost $f(M)$ and move it from the list VALID to the list NON-VALID.
- Step 4. If M is the final marking, construct the searched way from the initial marking to the final marking, and finish.
- Step 5. Calculate $v_d(t=j)$ (see section 3.2 - relation (3.1)) and generate the successor markings for each enabled transition, and set the way from successors to M .
- Step 6. For each successor marking M' , do the following:
 - 6.1. If marking M' is not already on list VALID or list NON-VALID, then put M' on list VALID;
 - 6.2. If marking M' is already on list VALID and a way with a higher benefit is found, then direct its pointer along the current way;
 - 6.3. If marking M' is already on list NON-VALID and a way with a higher benefit is found, then direct its pointer along the current way and move M' from list NON-VALID to the list VALID.
- Step 7. Go to step 2.

For an assembly plan with n operations the complexity of this algorithm is $O(bn)$, where b is the capacity of list VALID. The algorithm complexity also depends upon the total number of nodes in ASPN as well as the total number of raw materials to perform the assembly operations concurrently [18-21]. For the example in Figure 3.8 the optimal assembly plan involves the transition (t_2, t_4, t_7, t_9) . This way may be updated in accordance to the values v_{di} , and v_{pj} , where $i = 1, \dots, 11$, and $j = 1, \dots, 9$, and to the marking of places TD_k , where $k = 1, \dots, 5$.

3.5 Conclusion

The approach presented in this paper is suited to distributed and asynchronous systems, such as construction ones, in which no global state and no global time is available, and therefore a partial order model is considered. This work proposes a methodology for design and implementation of adaptive assembly systems. In order to model the planning problem, assembly supervisor Petri nets (ASPN's) are introduced, with two functions: one attached to places and the other to transitions. The first is a resource utility function, which represents the value of a subassembly or a part to be

used, and the second function represents the cost of performing a particular assembly operation. A bottom-up state-space size estimation technique for ASPN has been described. The estimation relies on the computation of state estimating functions (Se-functions). A lot of model analysis and control algorithms are based on the model state-space, and there it is proven that they are affected by large state-space sizes. The benefits of this approach include the simple representation of Se-functions that facilitates automation, and the possibility to interject hand-computed results into the estimation. Errors in the estimation may result from changes in the ASPN model to permit analysis in the Se functions.

Further research will include the colored Petri nets into the ASPN's, and the calculus of their Se-functions. To incorporate the uncertainty caused by different assembly conditions and the quality of resources, a probability value is assigned to each transition. Probability values can be updated during process execution. Future research will also focus on adapting interpreted Petri nets to the presented approach.

References

1. Sampath M, Sengupta R, Lafortune S, Sinnamohideen K, Tewneketzis D (1995) Diagnosability of discrete event systems. *IEEE Trans. on Autom. Contr.* 40(9):1555-1575
2. Jalcin A, Boucher ThO (1999) An architecture for flexible manufacturing cells with alternate machining and alternate sequencing. *IEEE Trans. Robot. Autom.* 15(6):1126-1130
3. Liu Z, Towsley D (1994) Stochastic scheduling in tree-networks. *Adv. Appl. Probability* 20:222-241
4. Lahmar M, Ergan H, Benjafaar S (2003) Resequencing and feature assignment on an automated assembly line. *IEEE Trans. on Rob. and Autom.* 19(1):89-102
5. Viswandham N, Narahari Y, Johnson TL (1990) Deadlock prevention and deadlock avoidance in flexible manufacturing systems using petri net models. *IEEE Trans. Robotics Automat.* 6(6):438-456
6. Nebel B, Koehler J (1995) Plan reuse versus plan generation - A theoretical and empirical analysis. *Artif Intell.* 76:427-454
7. Suzuki T, Kanehara T, Inaba A, Okuma S (1993) On algebraic and graph structural properties of assembly petri net. In: *Proc. IEEE Int. Conf. Robot. Automat., Atlanta, G. A.*, pp 507-514
8. Zussman E, Zhou MCh (1999) A methodology for modelling and adaptive planning of disassembly processes. *IEEE Trans. on Robot. and Autom.* 15(1):190-193

9. Zussman E, Zhou MCh (2000) Design and implementation of an adaptive process planner for disassembly processes. *IEEE Trans. on Robot. and Autom.* 16(2):171-179
10. Rogers RV, Preston White jr. K (1991) Algebraic mathematical programming and network models of the deterministic job-shop scheduling problem. *IEEE Trans. Syst. Man. and Cybern.* 21(3):350-361
11. Xie X (2002) Fluid stochastic event graph for evaluation and optimization of discrete event systems with failures. *IEEE Trans. on Robotics and Automation* 18(3):360-367
12. Watson JF, Desrochers AA (1992) Methods for estimating state-space size of Petri nets. In: *Proc. of the IEEE Int. Conf. Robotics Autom.*, Nice, France, pp 1031-1036
13. Ciufudean C (2001) Petri nets in railway traffic systems. *Advances in electrical and computer engineering*, University "Stefan cel Mare" of Suceava, vol 1(18), no 1(15):21-26
14. Ciufudean C (2002) Discrete event systems for modelling the railway traffic. Matrix Rom Publishing House, Bucuresti, Romania
15. Ciufudean C, Larionescu AB (2002) Safety criteria for production lines modeled with Petri nets. *Advances in Electrical and Computer Engineering*, vol 2(9), no 2(18):15-20
16. Ciufudean C (2007) Discrete event systems – Applied themes. Matrix Rom Publishing House, Bucuresti, Romania
17. Ciufudean C, Graur A, Filote C, Turcu C, Popa V (2006) Diagnosis of complex systems using ant colony decision petri nets. In: *First European Conference on Availability, Reliability and Diagnosis (ARES 2006)*, Wien, Austria, pp 567-573, ISBN 0-7695-2567-9
18. Ciufudean C, Filote C, Popescu D (2006) Workflows in constructions modelled with stochastic artificial Petri nets. In: *Proc. of The 23rd International Symposium on Automation and Robotics in Construction (ISARC 2006)*, Tokyo, Japan, pp 773-778, ISBN 4-9902717-1-8
19. Ciufudean C, Satco B, Filote C (2007) Reliability Markov chains for security data transmitter analysis. In: *Proc. of The Second International Conference on Availability, Reliability and Security (ARES 2007)*, Vienna University of Technology, Austria, pp 886-892, ISBN 0-7695-2775-2
20. Ciufudean C, Satco B, Nitu C (2007) Markov decision models for immune mechanism analysis. *WSEAS Transaction on Information Science and Applications* 4(1):23-29, ISSN 1790-0832
21. Ciufudean C, Graur A, Filote C, Turcu C (2007) A new formalism for failure diagnosis: Ant colony decision Petri nets. *Journal of Software (JSW)* 2(1):39-46, ISSN: 1796-217X

Chapter 4

Open architecture systems for MERO walking robots control

Luige Vladareanu,¹ Radu Munteanu,² Adrian Curaj,³ Ion N. Ion³

¹Institute of Solid Mechanics, Romanian Academy, C-tin Mille 15, Bucharest, luigiv@arexim.ro,

²Rector, Tehnical University of Cluj Napoca

³POLITEHNICA University of Bucharest, ROMANIA

Abstract. The paper presents an Open Architecture system for the MERO walking modular robot, which is a robot on 3x6 degrees of freedom, autonomous, being able to interact with the environment. Starting from purpose kinematics and kinetostatics analysis the mathematic model of the inverted kinematics is determined for controlling the main trajectory that defines legs' support and weigh centre positions. Related to this there is presented an Open Architecture system for the MERO robot position control in Cartesian coordinates through real time processing of the Jacobean matrix obtained out of the forward kinematics using the Denavit-Hartenberg method and calculating the Jacobean inverted matrix for feedback. The obtained results prove a significant reduction of the execution time for the real time control of MERO robot's position in Cartesian coordinates and increased flexibility.

Keywords. Real time control, walking robots, open-architecture systems, multiprocessor systems

4.1 Introduction

A robot can be viewed as a serial link manipulator where the links sequence is connected by an actuated joints mathematical relation which ensures coordinate transformation from one axis to the other. In general, due to having six degrees of freedom, the mathematical analysis for the manipulator becomes very complicated. There are two dominant coordinates systems: Cartesian coordinates and joint coordinates. Joint coordinates represent angles between links and link extensions. They form the coordinates where the robot links are moving with direct control by the actuators. The position and orientation of each segment of the linkage structure can be described using the Denavit-Hartenberg [D-H] transformation [1,2].

Kinematics modeling. Kinematics modeling of the legs' mechanisms related to the main system attached to the MERO modular walking robot is presented in Figure 4.1. There are 3 modules with 2 legs each, connected by 3 joint angles. Each leg is composed of a mechanism with 3 elements connected by 3 rotation actuators. The **D-H** matrix method in homogenous coordinates is used in order to determine the position of the legs against the platform. Following the analysis of the **D-H** links system for the robot leg mechanism we have found a quite particular position [3], characterized by the following values: $\alpha_1 = 900$, $s_3 = 0$, $\alpha_2 = 0$, $a_1 = 0$, $\alpha_3 = 0$, $a_2 = 11 = 250$ mm, $s_1 = 30$ mm, $a_3 = 12 = 250$ mm, $s_2 = 0$. This led to decreasing the number of parameters of the transformation matrix A_i from six to four.

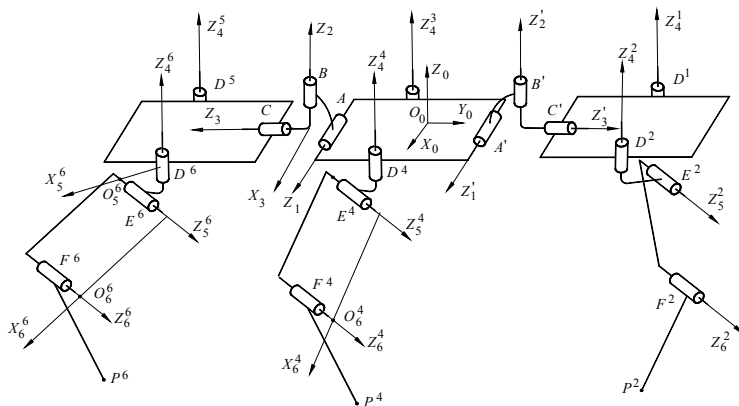


Fig. 4.1. The H-D axis system attached to the leg of MERO modular walking robot

The work presents both the transforming connections of the point's coordinates – from system **(i+1)** to the system **(i)**, that are realized by a transformation matrix A_i – as well as the transforming relations of the P_i

point's coordinates from the system attached to the element (3) of the leg in the platform system. Considering as already known the legs coordinates X_{OIP} , Y_{OIP} , Z_{OIP} , the mathematic model with variable parameters of the leg joint angles θ_{ij} ($j=1,6$, $i=4,6$) has been determined by inverted kinematics analysis. Thus, knowing the trajectory that the legs should describe, the parameters necessary to the control system in real time of the robot can be established. Using direct kinematics analysis are calculated X_{OIP} , Y_{OIP} , Z_{OIP} , and by inverted kinematics analysis θ_{4i} , θ_{5i} , θ_{6i} . The relative speed is calculated by derivation of movement equation and the relative accelerations by double derivation.

In the general case of the walking robots with $2n$ legs – the “myriapod” robots having “ n ” connected modules – the legs kinematics related to the environment system have been determined by the same algorithm.

4.2 Robot position control

The control using forward kinematics consists of transforming the actual joint coordinates, resulting from transducers, to Cartesian coordinates and compared with the desired Cartesian coordinates. The resulting error to the reference position in Cartesian coordinated is transmitted to every axis processing system. Using the inverting Jacobean matrix the angular errors will be generated on every motor control axis [3, 4]. There were identified five main processes for robot control: process 1-current position determined by the matrix in X_C Cartesian coordinates; process 2-error position resulting in δX_C Cartesian coordinates; process 3-Jacobean matrix setting; process 4-obtaining the triangular Jacobean matrix; process 5- processing joint errors $\delta \theta_{i \ 1,2,3}$, $i=1-6$, by inverse substitution.

The robot position control based on the **D-H** transformation is illustrated in Figure 4.2. The robot joint angles, θ_c , are transformed in X_c - Cartesian coordinates with **D-H** transformation, where a matrix results from X_C and $\delta \theta$, as below, with θ_j -joint angular j -offset distance, a_j - link length, α_j - twist.

Position and orientation of the end effectors with respect to the base coordinate frame is given by $X_C = A_1 * A_2 * A_3$. Position error ΔX is obtained as a difference between desired and current position. There is difficulty in controlling robot trajectory, if the desired conditions are specified using position difference ΔX with continuous measurement of $\delta \theta_{1,2,3}$, current position. The relation between the end-effector's position and orientation considered in Cartesian coordinates given at a certain moment and the robot joint angles $\delta \theta_{1,2,3}$, is : $x_i = f_i(\theta)$, where θ is the vector representing

the freedom degrees of robot axis. By derivation we will obtain: $\delta^3 \mathbf{X}_3 = \mathbf{J}(\theta) * \delta \theta_{1,2,3}$, where $\delta^3 \mathbf{X}_3$ represents differential linear and angular changes in the end effectors at the currently values of $\mathbf{X}_3$ and $\delta \theta_{1,2,3}$ representing the differential change of the set of joint angles. $\mathbf{J}(\theta)$ is the Jacobean matrix in which the elements $\mathbf{a}_{ij}$ satisfy the relation: $\mathbf{a}_{ij} = \delta \mathbf{f}_{i-1} / \delta \theta_{j-1}$, where i, j correspond to the dimensions of $\mathbf{x}$ respectively θ . The inverse Jacobean transforms the Cartesian position $\delta^3 \mathbf{X}_3$ respectively $\Delta \mathbf{X}$ into joint angular errors ($\Delta \theta$): $\delta \theta_{1,2,3} = \mathbf{J}^{-1}(\theta) * \delta^3 \mathbf{X}_3$.

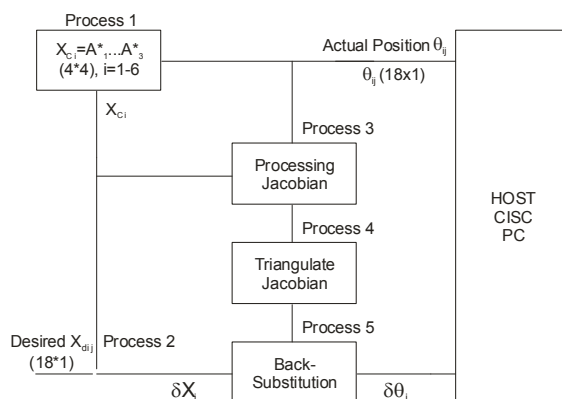


Fig. 4.2. The robot position control based on the **D-H** transformation

The Jacobean calculation consists in consecutive multiplication of the $\mathbf{A}$ manipulator matrix. The Gaussian elimination method provides an efficient implementation of the matrix inversion. The method consists of reducing the $\mathbf{J}$ matrix to the upper triangulate form and finding errors in $\Delta \theta$ joint coordinates using back-substitution. The joint angular errors $\Delta \theta$ can be used directly as control signals for robot motors.

The Multiprocessor System PLC. As a result of the studies, for the position control of the MERO walking robot a real time control system with open architecture OAH has been conceived. The system has four main components: the programmable logical controller (PLC0), PC system with open architecture (PC-OPEN), multiprocessor system PLC (SM-PLC) and the position control system (CP^k_i , $i=1-3$, $k=1-6$). The system ensures the implementation of the algorithm for robot position control in Cartesian coordinates through real time processing of the Jacobean matrix obtained by direct kinematics of the robot, respectively of the invert Jacobean matrix for feedback [4]. Joint angular errors for actuators on each robot freedom axis corresponding to the robot's new desired position generated in Cartesian coordinates are obtained by processing the Jacobean and inverted substitution.

The Multiprocessor System PLC (SM-PLC) and PLC0 has the role to send in real time, through the ARCNET fast communication network, the angular reference positions for the position regulator. For feedback, the current values θ^k_{ci} , ($i=1-3$, $k=1-6$), received from the position transducers (TP_{ki} , $i=1-3$, $k=1-6$) are transmitted by PLC0 through ARCNET. Five processes for implementing the mathematical model of robot control have been identified. In the active topology for process (1), each PLC generates an ascendant data flux from PLC0 to PLC5.

By calculating the transformed matrix ${}^iA^k_j$ for the leg k ($k=1-6$), from the axis i to the axis j , we obtain the coordination matrix in axis j , and finally the resulting coordinates for the robot environment $X^k_c = {}^1A^k_3 = A_1 * A_2 * A_3$. In the active topology for the process (2), the matrix ${}^{i-1}A^k_i$ is stored for each PLC, the Cartesian coordinates X^k_i in i axis, by multiplying with ${}^1A^k_{j-1}$, are determined and the position variation δX^k_c is calculated. The Jacobean matrix is obtained during process (3) by an ascending data flow, correlated with process (1). During the matrix calculation ${}^1A^k_j$, the PLC0 is assigned to $J(\theta_1)$, matrix of (3×1) , respectively PLC5 to $J(\theta_3)$.

In a process sequential with process (4) takes place the active process (5) which determines by inverted substitution the $\delta\theta$; angular error value in accordance to the relation: $\delta\theta_{1,2,3} = J^{-1}(\theta) * \delta^3 X_3$. To each PLC there has been allotted a column of the Jacobean matrix, the data flux going from PLC0 to PLC5 for triangulation and from PLC5 to PLC0 for inverted substitution. With the help of the relations from the mathematical model the execution program for the PLC0-PLC5 has been conceived and executed, in which each central unit has the role of MASTER communication by data flux through the ARCNET network (Figure 4.3).

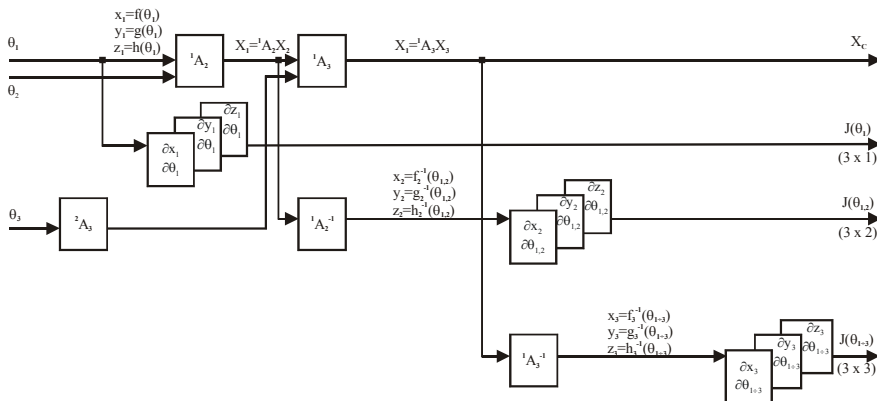


Fig. 4.3. The data flux for OAH implementation with PLC multiprocessor systems.

The active process (5) takes place within a sequential process with process (4), determining the value of the angular error $\delta\theta_i^k$ by inverted substitution.

4.3 Conclusions and results

Mathematical modeling of the movement of the MERO walking robot has led to the development of two applications. For the transportation of nuclear material (Figure 4.4) the imposed requirement is that the transported load maintain the same coordinate of the Z axis (vertical axis) throughout the movement.

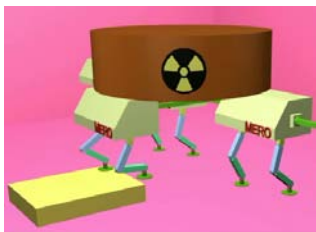


Fig. 4.4. MERO modular walking robot for Nuclear Transport

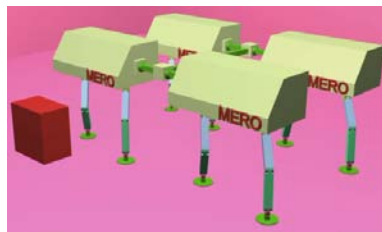


Fig. 4.5. MERO walking robot with detour around obstacles

It is of note that after reaching an obstacle, the forward legs have a different movement from the rest so that the height of the transported load is constant. For a walking robot which moves in the presence of obstacles which need to be circumvented (Figure 4.5), when reaching the object the robot changes direction by successive rotation of all legs. The results of this research have been materialized through MERO modular walking robot (Figure 4.6), which was built and tested at “Politehnica” University of Bucharest. Owing to the great computation speed of microprocessor systems and serial connection links for data transmission, the time necessary for establishing the inverted matrix is short enough to allow the robot control in real time, with no influence in performing the other programs.

The results show that the time necessary to perform the program for the MERO robot position control in Cartesian coordinates is 80% shorter. Moreover, the shorter execution time will ensure a faster feedback, allowing other programs to be performed in real time as well, such as the prehension force control, objects recognition, making it possible for the PC-OHA system to have a flexible and friendly human interface.

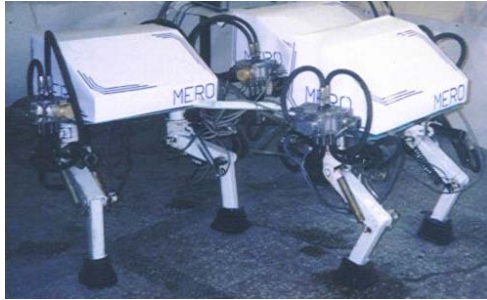


Fig. 4.6. MERO modular walking robot

References

1. Denavit J, Hartenberg RS (1955) A kinematic notation for lower-pair mechanism based on matrices. *ASME Journal of Applied Mechanics* 22(2):215-221
2. Hiller M, Muller J, Roll U, et al. (1999) Design and realization of the anthropomorphically legged and wheeled duisburg robot alduro. In: *The Tenth World Congress on the Theory of Machines and Mechanisms*, Oulu, Finland
3. Sidhu GS (1997) Scheduling algorithm for multiprocessor robot arm control. In: *Proc. 19th Southeastern Symp*
4. Vladareanu L, Peterson T (2003) New concepts for the real time control of robots by open architecture systems. *Machine Building* 55(11), ISSN 0573-7419

Chapter 5

Mass transfer analysis in the case of the EFG method

Liliana Braescu,¹ Thomas F. George,² Stefan Balint³

¹Department of Computer Science, West University of Timisoara, Blv. V. Parvan 4, Timisoara 300223, Romania

²Office of the Chancellor and Center for Nanoscience, Departments of Chemistry & Biochemistry and Physics & Astronomy, University of Missouri-St. Louis, St. Louis, MO 63121, USA

Abstract. The dependence of the Nd^{3+} ion distribution on the pulling rate v_0 and on the capillary channel radius R_{cap} , in an YVO_4 cylindrical bar grown from the melt by the edge-defined film-fed growth (EFG) method with central capillary channel (CCC) shaper and melt replenishment is analyzed. The partial differential equations used for describing the growth process are those of the incompressible Navier-Stokes equations in the Boussinesq approximation and of the convection-conduction and conservative convection-diffusion equations. For a numerical solution, the finite-element numerical technique using COMSOL Multiphysics 3.3 software for a 2D axis-symmetric model is utilized. The Marangoni effect (i.e., surface tension driven flows due to the temperature gradient along the liquid free surface) is taken into account, and for its implementation the weak form of the boundary application mode is employed. The computations are carried out in the stationary case, for different pulling rates and capillary channel radii. The numerical results are compared with the experimental data (Bunoiu *et al.* 2005).

Keywords. Edge-defined film-fed growth, Single crystal growth, Fiber, Rare earth compounds, Mass transfer

5.1 Introduction

Single crystals of rare-earth vanadates possess various useful properties for laser and optical applications (Kochurikhin *et al.* 2001; Hur *et al.* 2002; Shur *et al.* 2004). Nd:YVO₄ is one of the most efficient laser host crystals currently existing for diode-pumped solid state lasers.

Nd:YVO₄ single crystals were produced by many growth techniques (Czochralski, Verneuil, floating zone, laser-heated pedestal growth, top-seeded solution growth). Wide-range applications are still limited due to the serious difficulties in growing crystals with good quality; the homogeneity of the dopant concentration and free defects in the grown crystal are very important for optical applications. For solving the non-homogeneity problem, the edge-defined film-fed growth (EFG) method has been considered as a prospective way to attain high-quality rare-earth vanadate single crystals.

In this paper, the Nd³⁺ ion distribution is analyzed numerically in the framework of a stationary model, including incompressible fluid flow in the Boussinesq approximation, heat and mass transfer, and the Marangoni effect. The computations are made by using the 2D axis-symmetric hypothesis, for a cylindrical bar of radius $R = 1.5 \times 10^{-3}$ m grown with a pulling rate v_0 in the range 10^{-7} - 10^{-5} m/s, using a meniscus height (liquid free surface) of $h = 0.5 \times 10^{-3}$ m. An EFG growth system with a central capillary channel shaper design (Bunoiu *et al.* 2005; Braescu *et al.* 2006; George *et al.* 2007) and a die radius $R_0 = 2 \times 10^{-3}$ m is considered for five values of the capillary channel radius $R_{cap} = 0.5, 0.75, 1, 1.25, 1.5 \times 10^{-3}$ m; it is assumed that the melt level in the crucible is constant (continuous melt replenishment).

5.2 Problem formulation

In order to find the stationary impurity distribution in the crucible, capillary channel and in the meniscus, the stationary incompressible Navier-Stokes in Boussinesq approximation (the temperature-dependent density appears only in the gravitational force term), stationary convection-conduction and stationary conservative convection-diffusion equations are considered:

$$\left\{ \begin{array}{l} \rho_l (\bar{\mathbf{u}} \cdot \nabla) \bar{\mathbf{u}} = \nabla \cdot \left[-p \mathbf{I} + \eta \left(\nabla \bar{\mathbf{u}} + (\nabla \bar{\mathbf{u}})^T \right) \right] + \beta \rho_l \bar{\mathbf{g}} \cdot (T - \Delta T) \\ \nabla \cdot \bar{\mathbf{u}} = 0 \\ \nabla \cdot (-k \nabla T + \rho_l C_p T \bar{\mathbf{u}}) = 0 \\ \nabla \cdot (-D \nabla c + c \bar{\mathbf{u}}) = 0 \end{array} \right. \quad (5.1)$$

for which axisymmetric solutions are searched in the cylindrical-polar coordinate system (rOz) (see Figure 5.1). In the above system, the unknowns are: velocity vector $\bar{\mathbf{u}} = (u, v)$, the temperature T , impurity concentration c , pressure p . The material parameters are: ρ_l - reference melt density, β - heat expansion coefficient, η - dynamical viscosity, k - thermal conductivity, C_p - heat capacity, D - impurity diffusion, and $\bar{\mathbf{g}}$ - gravitational acceleration. The system (5.1) is considered in the two-dimensional domain whose boundary is from Ω_1 to Ω_{10} , as represented on Figure 5.1(a).

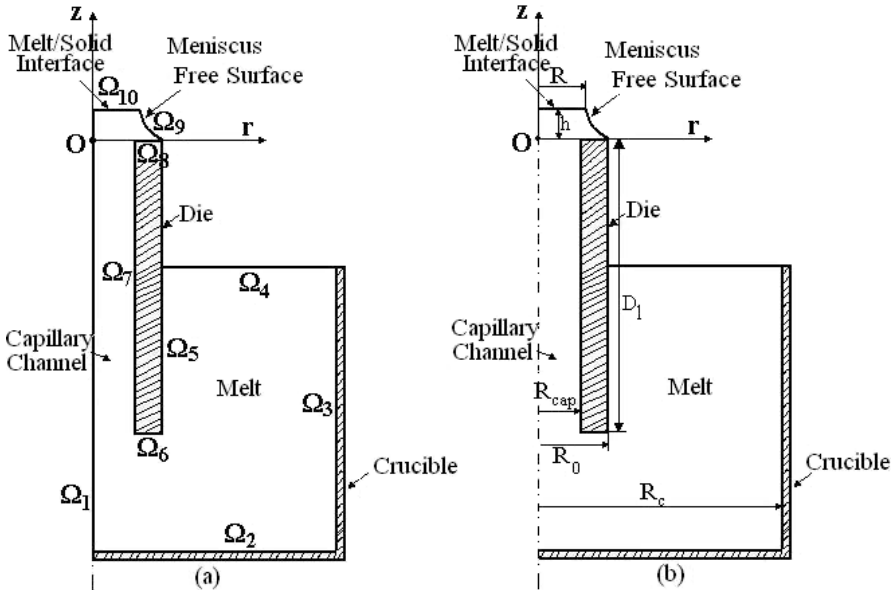


Fig. 5.1. Schematic diagram of the EFG system and boundary regions used in the numerical model

The fluid density is assumed to vary with temperature as:

$$\rho(r, z) = \rho_l [1 - \beta(T(r, z) - \Delta T)] \quad (5.2)$$

and the surface tension γ in the meniscus is assumed to vary linearly with temperature as:

$$\gamma(r, z) = \gamma_l + \frac{d\gamma}{dT}(T(r, z) - \Delta T) \quad (5.3)$$

where $\Delta T = (T_0 + T_m)/2$ is the reference temperature at the free surface, γ_l is the surface tension at the temperature ΔT and $d\gamma/dT$ is the rate of change of surface tension with the temperature.

For solving the system (5.1), boundary conditions on Ω_1 to Ω_{10} are imposed, with the Oz -axis being considered as a line of symmetry for all field variables:

- Flow conditions: On the melt/solid interface, the condition of outflow velocity is imposed, i.e., $\bar{\mathbf{u}} = v_0 \bar{\mathbf{k}}$, where $\bar{\mathbf{k}}$ represents the unit vector of the Oz -axis. On the melt level in the crucible, the inflow velocity condition is imposed (melt replenishment), $\bar{\mathbf{u}} = -\frac{\rho_s}{\rho_l} \cdot \frac{R^2}{R_c^2 - R_0^2} \cdot v_0 \bar{\mathbf{k}}$. On the meniscus free surface, we set up the slip/symmetry condition, $\bar{\mathbf{t}} \cdot \left[-p\mathbf{I} + \eta \left(\nabla \bar{\mathbf{u}} + (\nabla \bar{\mathbf{u}})^T \right) \right] \bar{\mathbf{n}} = 0$, $\bar{\mathbf{u}} \cdot \bar{\mathbf{n}} = 0$, where $\bar{\mathbf{t}}$ and $\bar{\mathbf{n}}$ represent the tangent and normal vectors, respectively. The other boundaries are setup by the non-slip condition $\bar{\mathbf{u}} = 0$.
- Thermal conditions: On the melt/solid interface, we set up the temperature $T = T_m$. On the free surface, we impose thermal insulation, i.e., $\bar{\mathbf{n}} \cdot (-k\nabla T + \rho_l C_p T \bar{\mathbf{u}}) = 0$. For the other boundaries, we have the temperature condition $T = T_0$.
- Concentration conditions: On the melt/solid interface the flux condition is imposed, which expresses that impurities are rejected into the melt according to $\frac{\partial c}{\partial \mathbf{n}} = -\frac{v_0}{D}(1 - K_0)c$. On the melt level in the crucible, we establish the concentration of Nd in alloy as $c = C_0 = 2$ at. % = 6.842×10^{-22} mol/m³. The other boundaries are setup by insulation/symmetry, $\bar{\mathbf{n}} \cdot (-D\nabla c + c\bar{\mathbf{u}}) = 0$.

Together with these boundary conditions, on the free surface the Marangoni effect is employed by using the weak form of the boundary application mode. Thus, on the free surface (meniscus), we impose the condition which expresses that the gradient velocity field along the meniscus is balanced by the shear stress:

$$\eta \cdot [t_r \quad t_z] \cdot \begin{bmatrix} 2 \frac{\partial u}{\partial r} & \frac{\partial u}{\partial z} + \frac{\partial v}{\partial r} \\ \frac{\partial u}{\partial z} + \frac{\partial v}{\partial r} & 2 \frac{\partial v}{\partial r} \end{bmatrix} \cdot \begin{bmatrix} n_r \\ n_z \end{bmatrix} = \frac{d\gamma}{dT} \cdot \left(\frac{\partial T}{\partial r} + \frac{\partial T}{\partial z} \right) \quad (5.4)$$

where $\bar{t} = (t_r, t_z)$ and $\bar{n} = (n_r, n_z)$. The sign of the rate $d\gamma/dT$, in general, depends on the material, with leading outward ($d\gamma/dT < 0$) or inward ($d\gamma/dT > 0$) flow along the liquid surface.

5.3 Numerical results

The computations are made in the axis-symmetric domain (Figure 5.1) in the stationary case, for a Nd doped YVO₄ cylindrical bar having a radius of $R = 1.5 \times 10^{-3}$ m, grown with a pulling rate v_0 in the range 10^{-7} - 10^{-5} m/s. The boundaries presented in Figure 5.1 are determined from the particularities (characteristic elements) of the considered EFG growth system. Thus, a crucible with inner radius $R_c = 23 \times 10^{-3}$ m is considered, which is continuously fed with Nd-doped melted YVO₄ such that the melt height in the crucible is maintained constant at 45×10^{-3} m. In the crucible, a die with the radius of $R_0 = 2 \times 10^{-3}$ m and length was introduced, such that 2/3 of the die 45×10^{-3} m is immersed in the melt. In the die, a capillary channel is manufactured through which the melt climbs to the top of the die, where a low meniscus of height $h = 0.5 \times 10^{-3}$ m is formed. In the computations, five different radii of the capillary channel are considered: $R_{cap} = 0.5, 0.75, 1, 1.25, 1.5 \times 10^{-3}$ m.

The material parameters used in the numerical simulations and their significance for Nd:YVO₄ are given in Table 5.1 (Kochurikhin *et al.* 2001; Hur *et al.* 2002; Shur *et al.* 2004; Kochurikhin 2007).

The employed mesh is considered with maximum element size of $1e-3$ and manually refined along the boundaries 8, 9 and 10 (maximum element size $1e-5$). Thus, according to the considered geometries corresponding to $R_{cap} = 0.5, 0.75, 1, 1.25, 1.5 \times 10^{-3}$ m, we have 4823, 4589, 4231, 4153, and 3779 triangular elements. The stationary incompressible Navier-Stokes model for fluid flow, stationary heat transfer and mass transfer, and the weak form of the boundary condition imply for these meshes a total number of degrees of freedom equal to 44267, 41999, 38740, 37951, and 34592, respectively.

The computed fluid flow in the meniscus, for three values of R_{cap} , is presented in Figure 5.2 for the pulling rate $v_0 = 1e-5$ m/s.

Table 5.1. Material parameters for Nd:YVO₄

Nomenclature	Value
β heat expansion coefficient (1/K)	4.43e-6
c impurity concentration (mol/m ³)	
C_0 alloy concentration (at %)	2
C_p heat capacity (J/Kg×K)	800
$d\gamma/dT$ rate change of the surface tension (N/ m×K)	-5e-5
D impurity diffusion (m ² /s)	2.4e-9
D_l die length (m)	45e-3
g gravitational acceleration (m/s ²)	9.81
h meniscus height (m)	0.5e-3
k thermal conductivity (W/m×K)	5.2
k_g vertical temp. gradient [K/m]	33000
K_0 partition coefficient	0.5
η dynamical viscosity (Kg/m×s)	0.046
p pressure (Pa)	
R crystal radius (m)	1.5e-3
R_{cap} capillary channel radius (m)	
R_c inner radius of the crucible (m)	23e-3
R_0 die radius (m)	2e-3
ρ_l density of the melt (Kg/m ³)	4200
ρ_s density of the crystal (Kg/m ³)	4200
$\underline{u}$ velocity vector	
v_0 pulling rate (m/s)	
T temperature (K)	
T_0 temperature at $z = h$ (K)	
T_m melting temperature (K)	2083
ΔT reference temperature (K)	
z coordinate in the pulling direction	

This reveals that for a fixed pulling rate, the flow velocity decreases if R_{cap} increases.

The maximum of the fluid flow is achieved on the meniscus free surface, and the arrows denoting the flow of the velocity field caused by the surface tension driven flow (outward flow because $d\gamma/dT < 0$) show that the Marangoni effect plays an important role in mass transfer, as it is reported by the crystal growers (Bunoiu *et al.* 2005). The impurities are moved from the lateral meniscus free surface, inside the meniscus near the melt/solid interface.

This combined effect of the buoyancy and Marangoni forces can be seen in Figures 5.3 and 5.4 for two values of R_{cap} .

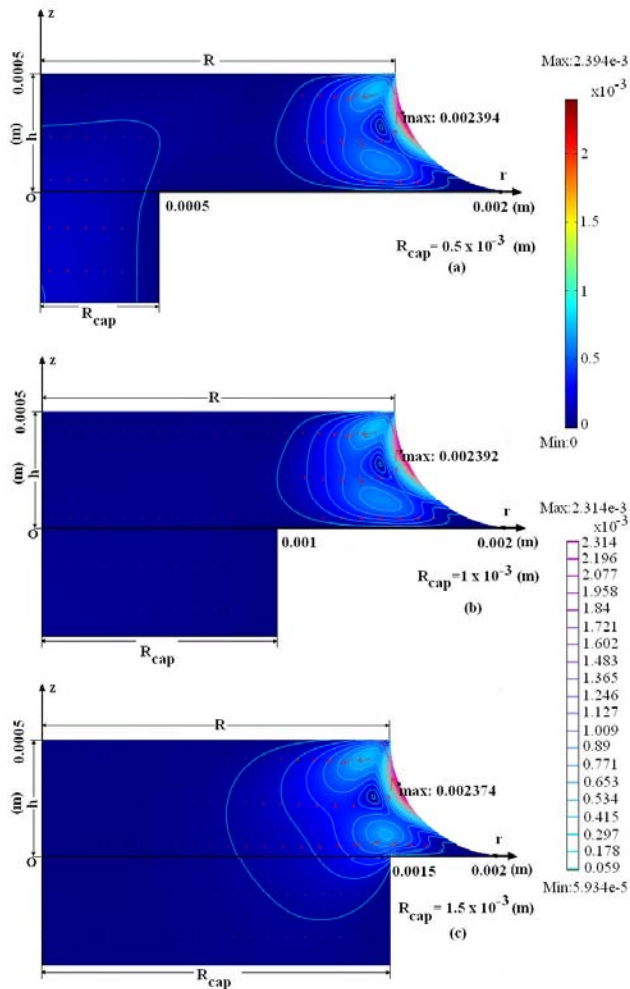


Fig. 5.2. Fluid flow field in the meniscus for $v_0 = 1e-5$ m/s

Computations show that for capillary radii less than the desired crystal radius, at the level $z = h$ (i.e., the melt/solid interface), the radial concentration increases, has a maximum, and after that decreases. The point where the radial concentration has a maximum depends on the pulling rate v_0 , and in general, it is located at a distance on the same order as the meniscus height from the external crystal surface. This behavior of the Nd^{3+} concentration is in agreement with experimental data reported for another oxide (sapphire) cylindrical bar of radius 2.25×10^{-3} m, grown in an EFG system with a CCC die of radius $R_0 = 2.35 \times 10^{-3}$ m and a capillary channel radius $R_{\text{cap}} = 0.5 \times 10^{-3}$ m.

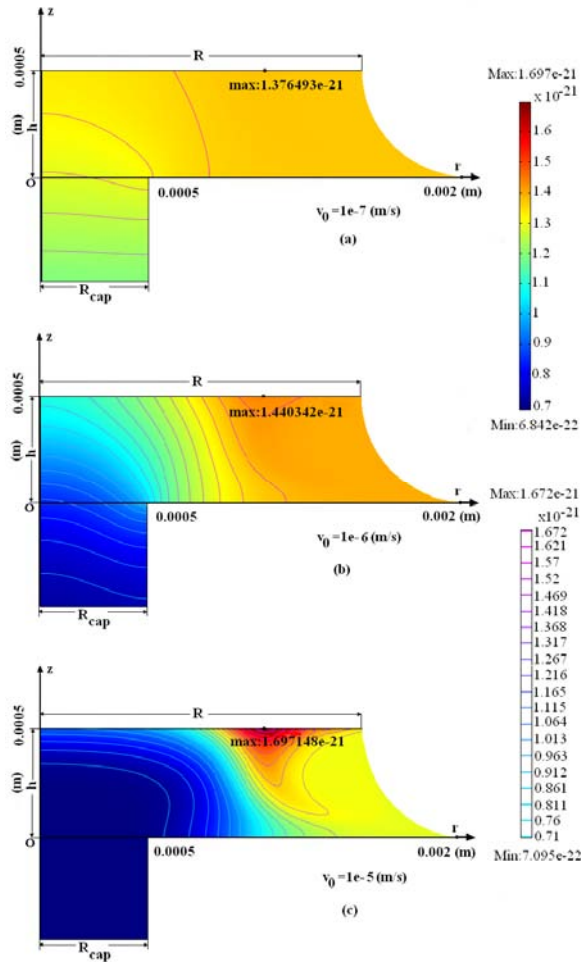


Fig. 5.3. Dependence of the Nd^{3+} ion distribution on the pulling rate for $R_{\text{cap}} = 0.5 \times 10^{-3} \text{ m}$.

Computations show that for capillary radii less than the desired crystal radius, at the level $z = h$ (i.e., the melt/solid interface), the radial concentration increases, has a maximum, and after that decreases. The point where the radial concentration has a maximum depends on the pulling rate v_0 , and in general, it is located at a distance on the same order as the meniscus height from the external crystal surface. This behavior of the Nd^{3+} concentration is in agreement with experimental data reported for another oxide (sapphire) cylindrical bar of radius $2.25 \times 10^{-3} \text{ m}$, grown in an EFG system with a CCC die of radius $R_0 = 2.35 \times 10^{-3} \text{ m}$ and a capillary channel radius $R_{\text{cap}} = 0.5 \times 10^{-3} \text{ m}$.

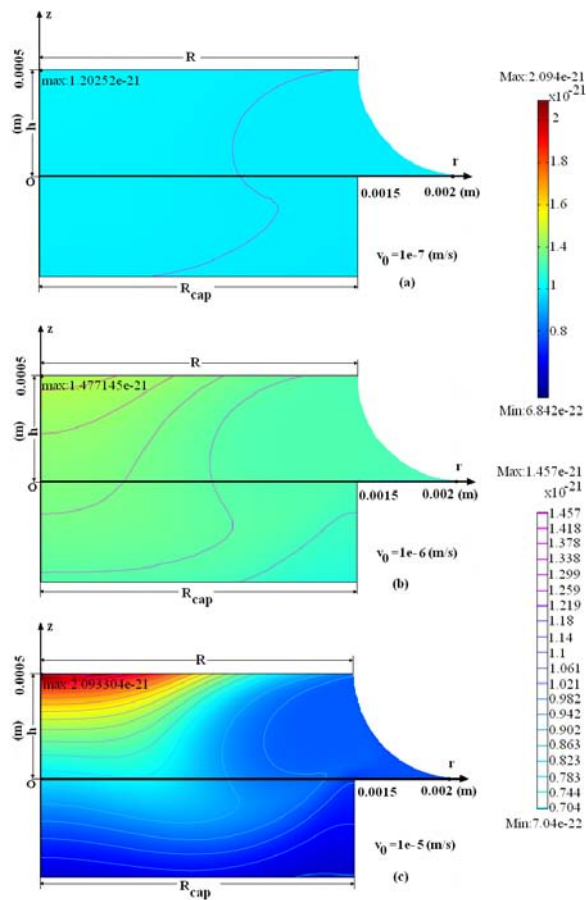


Fig. 5.4. Dependence of the Nd^{3+} ion distribution on the pulling rate for $R_{\text{cap}} = 1.5 \times 10^{-3} \text{ m}$

Values of the impurity concentration at different points on the melt/solid interface and the maximum values $c_{\text{max}}(r,z) = c_{\text{max}}(r,h)$ are presented in Tables 5.2-5.6.

Table 5.2. Nd^{3+} ion concentration on the melt/solid interface $z = h$ for $R_{\text{cap}} = 0.5 \times 10^{-3} \text{ m}$

$v_0(\text{m/s})$	1e-7	1e-6	1e-5
$(r;h)$			
(0;0.5e-3)	1.33e-21	1.09e-21	0.90e-21
(0.5e-3;0.5e-3)	1.35e-21	1.21e-21	0.96e-21
(1e-3;0.5e-3)	1.37e-21	1.43e-21	1.67e-21
(1.5e-3;0.5e-3)	1.37e-21	1.40e-21	1.34e-21

Table 5.3. Nd³⁺ ion concentration on the melt/solid interface $z = h$ for $R_{cap} = 0.75 \times 10^{-3}$ m

$v_0(\text{m/s})$	1e-7	1e-6	1e-5
$(r;h)$			
(0;0.5e-3)	1.35e-21	1.20e-21	0.99e-21
(0.5e-3;0.5e-3)	1.35e-21	1.26e-21	1.02e-21
(1e-3;0.5e-3)	1.37e-21	1.42e-21	1.65e-21
(1.5e-3;0.5e-3)	1.36e-21	1.38e-21	1.32e-21

Table 5.4. Nd³⁺ ion concentration on the melt/solid interface $z = h$ for $R_{cap} = 1.0 \times 10^{-3}$ m

$v_0(\text{m/s})$	1e-7	1e-6	1e-5
$(r;h)$			
(0;0.5e-3)	1.34e-21	1.26e-21	1.07e-21
(0.5e-3;0.5e-3)	1.34e-21	1.30e-21	1.11e-21
(1e-3;0.5e-3)	1.35e-21	1.40e-21	1.63e-21
(1.5e-3;0.5e-3)	1.35e-21	1.36e-21	1.27e-21

Table 5.5. Nd³⁺ ion concentration on the melt/solid interface $z = h$ for $R_{cap} = 1.25 \times 10^{-3}$ m

$v_0(\text{m/s})$	1e-7	1e-6	1e-5
$(r;h)$			
(0;0.5e-3)	1.27e-21	1.29e-21	1.11e-21
(0.5e-3;0.5e-3)	1.28e-21	1.32e-21	1.19e-21
(1e-3;0.5e-3)	1.29e-21	1.40e-21	1.61e-21
(1.5e-3;0.5e-3)	1.28e-21	1.36e-21	1.21e-21

Table 5.6. Maximum Nd³⁺ ion concentration $c_{max}(r,h)$ for different v_0 and R_{cap}

$v_0(\text{m/s})$	1e-7	1e-6	1e-5
R_{cap}			
0.5e-3	$c_{max}(0.00104,h)$ =1.3764e-21	$c_{max}(0.00104,h)$ =1.4403e-21	$c_{max}(0.00104,h)$ =1.6971e-21
0.75e-3	$c_{max}(0.00102,h)$ =1.3738e-21	$c_{max}(0.00103,h)$ =1.4226 e-21	$c_{max}(0.00104,h)$ =1.6740 e-21
1.0e-3	$c_{max}(0.00099,h)$ =1.3545e-21	$c_{max}(0.00099,h)$ =1.4084 e-21	$c_{max}(0.00100,h)$ =1.6384 e-21
1.25e-3	$c_{max}(0.00097,h)$ =1.2903 e-21	$c_{max}(0.00097,h)$ =1.4041 e-21	$c_{max}(0.00097,h)$ =1.6227 e-21

Computations show that the maximum Nd³⁺ ion concentration $c_{max}(r,h)$ decreases if R_{cap} increases or v_0 decreases.

For a capillary channel radius equal to the crystal radius (see Figure 5.4), the buoyancy force will move this maximum to the axis of the crystal. In this case, the radial concentration decreases from the crystal axis to the lateral surface as can be seen in Table 5.7.

Table 5.7. Nd^{3+} ion concentration on the melt/solid interface $z = h$ for $R_{cap} = 1.5 \times 10^{-3}$ m

$v_0(\text{m/s})$	1e-7	1e-6	1e-5
$(r;h)$			
(0;0.5e-3)	1.20e-21	1.47e-21	2.09e-21
(0.5e-3;0.5e-3)	1.19e-21	1.43e-21	1.89e-21
(1e-3;0.5e-3)	1.19e-21	1.35e-21	1.29e-21
(1.5e-3;0.5e-3)	1.18e-21	1.32e-21	1.01e-21

The Nd^{3+} concentration on the axis $C_{ax} = c(0,0.5e-3)$, on the lateral surface of the crystal $C_{lat} = c(1.5e-3,0.5e-3)$ and the difference between them depend on the pulling rate and on the capillary radius. A measure of the non-homogeneity of the crystal is given by the absolute value of the difference ($C_{lat}-C_{ax}$), called radial segregation. This segregation is small when the pulling rate is small and the capillary channel radius is large.

5.4 Conclusions

Mass transfer of Nd^{3+} in a YVO_4 cylindrical bar is analyzed in the framework of a stationary model solved numerical using the finite element method. In this model, the surface tension driven flows due to the temperature gradient along the liquid free surface is included. The impurity distribution in the melt and in the crystal is computed as function of the pulling rate v_0 and of the capillary channel radius R_{cap} . The dependence of the impurity distribution on v_0 and R_{cap} is established.

Acknowledgment. We are grateful to the North Atlantic Treaty Organisation (Grant CBP.EAP.CLG 982530) for support of this project.

References

1. Bunoiu O, Nicoara I, Santailler JL, Duffar T (2005) Fluid flow and solute segregation in EFG crystal growth process. *Journal of Crystal Growth* 275:e799-e805

2. Braescu L, Balint S, Tanasie L (2006) Numerical studies concerning the dependence of the impurity distribution on the pulling rate and on the radius of the capillary channel in the case of a thin rod grown from the melt by edge-defined film-fed growth (EFG) method. *Journal of Crystal Growth* 291:52-59
3. George TF, Balint S, Braescu L (2007) Mass and heat transport in Bridgman-Stockbarger and edge-defined film-fed growth systems. *Springer Handbook of Crystal Growth, Defects and Characterization*, Springer, Berlin Heidelberg New York
4. Hur MG, Yang WS, Suh SJ, Ivanov MA, Kochurikhin VV, Yoon DH (2002) Optical properties of EFG grown Nd:YVO₄ single crystals dependent on Nd concentration. *Journal of Crystal Growth* 237-239:745-748
5. Kochurikhin VV (2007) Personal communications. Head of Growth Team, General Physics Institute, Moscow, Russia, <http://webcenter.ru/~kochur/>
6. Kochurikhin VV, Ivanov MA, Yang WS, Suh SJ, Yoon DH (2001) Development of edge-defined film-fed growth for the production of YVO₄ single crystals with various shapes. *Journal of Crystal Growth* 229:179-183
7. Shur JW, Kochurikhin VV, Borisova AE, Ivanov MA, Yoon DH (2004) Photoluminescence properties of Nd:YVO₄ single crystals by multi- die EFG method. *Optical Materials* 26:347-350

Chapter 6

Information system architecture for data warehousing

Francisco Araque, Alberto Salguero, Cecilia Delgado

Dpto. De Lenguajes y Sistemas Informáticos,
University of Granada, 18071, Granada (Andalucía), Spain
E-mails: {faraque, agsh, cdelgado}@ugr.es

Abstract. One of the most complex issues of the integration and transformation interface is the case where there are multiple sources for a single data element in the enterprise Data Warehouse (DW). There are many facets due to the number of variables that are needed in the integration phase. However we are interested in the temporal integration problem. This paper presents our DW architecture for temporal integration on the basis of the temporal properties of the data and temporal characteristics of the data sources. The proposal shows the steps for the transformation of the native schemes of the data sources into the DW scheme. In addition we present two new components for temporal integration: Temporal Integration Processor and Refreshment Metadata Generator, which will be used for integrating temporal properties of data and generating the necessary data for the later refreshment of the DW.

Keywords. Data Warehouse, Information System, System Integration

6.1 Introduction

The ability to integrate data from a wide range of data sources is an important field of research in data engineering. Data integration is a prominent

theme in many areas and enables widely distributed, heterogeneous, dynamic collections of information sources to be accessed and handled.

Many information sources have their own information delivery schedules, whereby the data arrival time is either predetermined or predictable. If we use the data arrival properties of such underlying information sources, the Data Warehouse Administrator (DWA) can derive more appropriate rules and check the consistency of user requirements more accurately. The problem now facing the user is not the fact that the information being sought is unavailable, but rather that it is difficult to extract exactly what is needed from what is available.

It would therefore be extremely useful to have an approach which determines whether it would be possible to integrate data from two data sources (with their respective data extraction methods associated). In order to make this decision, we use the temporal properties of the data, the temporal characteristics of the data sources and their extraction methods. Notice that we are not suggesting a methodology, but an architecture. Defining a methodology is absolutely out of the scope of this paper, and the architecture does not impose it.

It should be pointed out that we are not interested in how semantically equivalent data from different data sources will be integrated. Our interest lies in knowing whether the data from different sources (specified by the DWA) can be integrated on the basis of the temporal characteristics (not in how this integration is carried out).

The use of DW and Data Integration has been proposed previously in many fields. In [1] the Integrating Heterogeneous Tourism Information data sources problem is addressed using three-tier architecture. In [2] a Real-Time Decision Support System for space missions control is put forward using Data Warehousing technology. In [3] a multi-level security policies integration methodology to endow tightly coupled federated database systems with a multilevel security system is presented. In [4] a framework for quality-oriented DW management is exposed, where special attention is paid to the treatment of metadata. The problem of the little support for automatized tasks in DW is considered in [5], where the DW is used in combination with event/condition/action rules to get an active DW. In [6] an integrated decision support system from the perspective of a DW is exposed. Their authors state that the essence of the data warehousing concept is the integration of data from disparate sources into one coherent repository of information. Nevertheless, none of the previous works encompass the aspects of the integration of the temporal parameters of data.

In this article a solution to this problem is proposed. Its main contributions are: a DW architecture for temporal integration on the basis of the temporal properties of the data and temporal characteristics of the sources,

a Temporal Integration Processor and a Refreshment Metadata Generator, that will be both used to integrate temporal properties of data and to generate the necessary data for the later DW refreshment.

The remaining part of this paper is organized as follows. In Section 6.2, the concept of DW and our previous related works are revised; in section 6.3 our architecture is presented; Finally, Section 6.4 summarizes the conclusions of this paper.

6.2 Federated databases and data warehouses

Inmon [7] defined a Data Warehouse as “a subject-oriented, integrated, time-variant, non-volatile collection of data in support of management’s decision-making process.” A DW is a database that stores a copy of operational data with an optimized structure for query and analysis. The scope is one of the issues which defines the DW: it is the entire enterprise. In terms of a more limited scope, a new concept is defined: a Data Mart (DM) is a highly focused DW covering a single department or subject area. The DW and data marts are usually implemented using relational databases, [8] which define multidimensional structures. A *federated database system* (FDBS) is formed by different *component database systems*; it provides integrated access to them: they co-operate (inter-operate) with each other to produce consolidated answers to the queries defined over the FDBS. Generally, the FDBS has no data of its own, queries are answered by accessing the component database systems.

We have extended the Sheth & Larson five-level architecture [9], which is very general and encompasses most of the previously existing architectures. In this architecture three types of data models are used: first, each component database can have its own native model; second, a *canonical data model* (CDM) which is adopted in the FDBS; and third, external schema can be defined in different *user models*.

One of the fundamental characteristics of a DW is its temporal dimension, so the scheme of the warehouse has to be able to reflect the temporal properties of the data. The extracting mechanisms of this kind of data from operational system will be also important. In order to carry out the integration process, it will be necessary to transfer the data of the data sources, probably specified in different data models, to a common data model, that will be the used as the model to design the scheme of the warehouse. In our case, we have decided to use an OO model as canonical data model, in particular, the object model proposed in the standard ODMG 3.0.

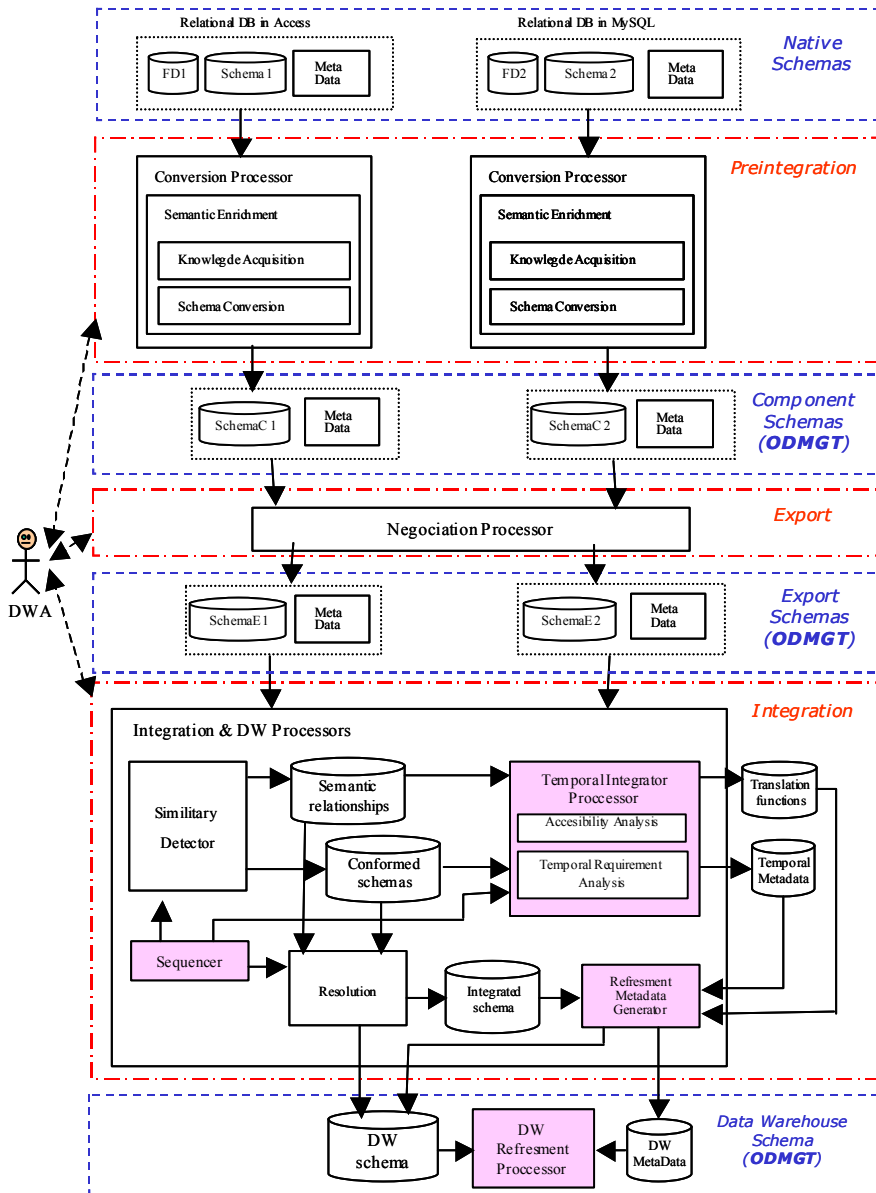


Fig. 6.1. Functional architecture

ODMG has been extended with temporal elements. We call this new ODMG extension as ODMGT. This is also our proposal: to use for the definition of the data ware-house and data mart schema an object-oriented

model as CDM, enhanced with temporal features to define loading of the data warehouse and data marts.

6.3 Architecture extension with temporal elements

Taking papers [9][10] as point of departure, we propose the following reference architecture (see Figure 6.1):

Native Schema. Initially we have the different data source schemes expressed in its native schemes. Each data source will have, a scheme, the data inherent to the source and the metadata of its scheme. In the metadata we will have huge temporal information about the source: temporal data on the scheme, metadata on availability of the source, availability of the log file or delta if it had them, etc.

Some of the temporal parameters, explained in depth in [11], that we consider of interest for the integration process are:

- *Availability Window (AW)*. Period of time in which the data source can be accessed by the monitoring programs responsible for data source extraction.
- *Extraction Time (ET)*. Period of time taken by the monitoring program to extract significant data from the source.
- *Granularity (Gr)*. It is the extent to which a system contains discrete components of ever-smaller size. In our case, because we are dealing with time, it is common to work with granules like minute, day, month...
- *Transaction time (TT)*. Time instant when the data element is recorded in the data source computer system. This would be the data source TT.
- *Storage time (ST)*. Maximum time interval for the delta file, log file, or a source image to be stored.

Preintegration. In the *Preintegration* phase, the semantic enrichment of the data source native schemes is made by the conversion processor. In addition, the data source temporal metadata are used to enrich the data source scheme with temporal properties. We obtain the component scheme (CST) expressed in the CDM, in our case, ODMGT (ODMG enriched with temporal elements).

Component and Export Schemas. Apart from the five-scheme levels mentioned (Sheth and Larson 1990), three more different levels should be considered:

- *Component Scheme T (CST)*: the conversion of a Native Scheme to our CDM, enriched so that temporal concepts could be expressed.

- *Exportation Scheme T (EST)*: it represents the part of a component scheme which is available for the DW designer. It is expressed in the same CDM as the Component Scheme.
- *Data Warehouse Scheme*: it corresponds to the integration of multiple Exportation Schemes T according to the design needs expressed in an enriched CDM so that temporal concepts could be expressed.

From the CST expressed in ODMGT, the negotiation processor generates the export schemes (EST) expressed in ODMGT. These EST are the part of the CST that is considered necessary for its integration in the DW.

Integration. From many data sources EST schemas, the DW scheme is constructed (expressed in ODMGT). This process is made by the Integration Processor that suggests how to integrate the Export Schemes helping to solve semantic heterogeneities (out of the scope of this paper). In the definition of the DW scheme, the DW Processor participates in order to contemplate the characteristics of structuring and storage of the data in the DW.

Two modules have been added to the reference architecture in order to carry out the integration of the temporal properties of data, considering the extraction method used: the *Temporal Integration Processor* and the *Metadata Refreshment Generator*.

The *Temporal Integration Processor* uses the set of semantic relations and the conformed schemes obtained during the detection phase of similarities [12]. This phase is part of the integration methodology of data schemes. As a result, we obtain data in form of rules about the integration possibilities existing between the originating data from the data sources (minimum granularity, if the period of refreshment must be annotated between some concrete values...). This information is kept in the Temporal Metadata Warehouse. In addition, as a result of the Temporal Integration process, a set of mapping functions is obtained. It identifies the attributes of the schemes of the data sources that are self-integrated to obtain an attribute of the DW scheme.

The *Metadata Refreshment Generator* determines the most suitable parameters to carry out the refreshment of data in the DW scheme [13]. The DW scheme is generated in the resolution phase of the methodology of integration of schemes of data. It is in this second phase where, from the minimum requirements generated by the temporal integration and stored in the Temporal Metadata warehouse, the DW designer fixes the refreshment parameters. As result, the DW scheme is obtained along with the Refreshment Metadata necessary to update the former according to the data extraction method and other temporal properties of a concrete data source.

Obtaining of the DW scheme and the Export schemes is not a linear process. We need the Integration and Negotiation Processors collaborate in an iterative process where the participation of the local and DW administrators is necessary [12].

Data Warehouse Refreshment. After temporal integration and once the DW scheme is obtained, its maintenance and update will be necessary. This function is carried out by the DW Refreshment Processor. Taking both the minimum requirements that are due to fulfill the requirements to carry out integration between two data of different data sources (obtained by means of the Temporal Integration module) and the integrated scheme (obtained by the resolution module) the refreshment parameters of the data stored in the DW will be adjusted.

6.4 Conclusion

In this paper we have presented a DW architecture for the temporal integration on the basis of the temporal properties of the data and temporal characteristics of the data sources. We have described the new modules introduced to the architecture of reference: Temporal Integration Processor and Refreshment Metadata Generator. These modules are responsible of checking the temporal parameters of data sources and of determining the best refreshment parameters according to the DWA requirements, generating, thus, the necessary data for the later DW refreshment. We used ODMGT (an ODMG extension with temporal elements) as CDM. This work has been supported by the  Research Program under project GR2007/07-2 and by the Spanish Research Program under projects EA-2007-0228 and TIN2005-09098-C05-03.

References

1. Haller M, Pröll B, Retschitzegger W, Tjoa AM, Wagner RR (2000) Integrating heterogeneous tourism information in TIScover - The MIRO-Web approach. In: Proceedings Information and Communication Technologies in Tourism, ENTER. Barcelona (España)
2. Moura Pires J, Pantoquilha M, Viana N (2004) Real-time decision support system for space missions control. In: International Conference on Information and Knowledge Engineering, Las Vegas, USA
3. Oliva M, Saltor F (2001) Integrating multilevel security policies in multilevel federated database systems. In: Thuraisingham B, van de Riet R, Dittrich KR,

- Tari Z (eds) *Data and Applications Security: Developments and Directions*, Kluwer Academic Publishers, Boston, pp 135–147
4. Vassiliadis P, Quix C, Vassiliou Y, Jarke M (2001) Data warehouse process management. *Information System* 26:205-236
5. Thalhammer T, Schrefl M, Mohania M (2001) Active data warehouses: complementing OLAP with analysis rules. *Data & Knowledge Engineering* 39(3):241-269
6. March ST, Hevner AR (2003) Integrated decision support: A data warehousing perspective. *AIS SIGDSS Pre-ICIS Workshop Research Directions on Decision Support*
7. Inmon WH (2002) *Building the data warehouse*. John Wiley
8. Harinarayan V, Rajaraman A, Ullman J (1996) Implementing data cubes efficiently. In: *Proc. of ACM SIGMOD Conference*, Montreal
9. Sheth A, Larson J (1990) Federated database systems for managing distributed, heterogeneous and autonomous databases. *ACM Computing Surveys*, 22(3)
10. Samos J, Saltor F, Sistac J, Bardés A (1998) Database architecture for data warehousing: An evolutionary approach. In: Quirchmayr G et al. (Eds.) *Proc. Int'l Conf. on Database and Expert Systems Applications*, Springer-Verlag, Vienna, pp 746-756
11. Araque F, Salguero AG, Delgado C, Garvi E, Samos J (2006) Algorithms for integrating temporal properties of data in Data Warehouse. In: *8th International Conference on Enterprise Information Systems (ICEIS)*, Paphos, Cyprus
12. Oliva M, Saltor F (1996) A negotiation process approach for building federated databases. In: *Proceedings of 10th ERCIM Database Research Group Workshop on Heterogeneous Information Management*, Prague, pp 43–49
13. Araque F, Samos J (2003) Data warehouse refreshment maintaining temporal consistency. In: *5th Intern. Conference on Enterprise Information Systems (ICEIS)*, Angers, France

Chapter 7

Dimensional dynamics identification of reconfigurable machine tools

Ciprian Cuzmin, Virgil Teodor, Nicolae Oancea, Vasile Marinescu, Alexandru Epureanu

Department of Manufacturing Engineering, “Dunărea de Jos” University of Galați, Galați, 111 Domnească street, zip 800201, ROMANIA, ciprian.cuzmin@ugal.ro

Abstract. The paper presents a new method of dimensional dynamics online identification to be used for the dimensional control of reconfigurable machining systems. This dimensional control is designed to be adaptive and predictive. The dimensional control of the reconfigurable machining system is a key action in achieving the desired quality of a final product. Since a decrease in deviation usually results in higher costs, the deviation compensation would be a better solution for improving the process accuracy. Deviation compensation occurring during the machining process requires a model able to describe the dimensional dynamics of a machining system, which stands for the relationship between the dimensional changing of part being machined and the process parameters. On the other hand the machining system behavior is changing significantly in time even when a small number of parts is processed. This is the reason why dimensional dynamics must reveal the changing in time of the parts dimensional variation- process parameters relationship.

Keywords. Reconfigurable machine tools, online identification, dimensional dynamics, dimensional control

7.1 Introduction

The dimensional control of a machining process is a key action to obtain the desired quality of a final product. Two techniques of dimensional deviation control are used:

- a. Diminishing the deviation by making the process parameters values smaller or using more consecutive passing;
- b. Deviation compensation by modifying the programmed tool-trajectory.

Since a decrease in deviation presents the major drawback of high costs, the deviation compensation seems to be a more suitable solution.

Deviation compensation requires deviation prediction based on a model able to describe the dimensional dynamics of the machining system ([1, 2, 4]). This dimensional dynamics is in fact the relationship between the dimension deviation of the part being machined and the processing parameters. On the other hand, the behavior of the machining system changes significantly in time, even when small part batches are processed.. This is the reason why dimensional dynamics must also take into account the changing in time of the relationship between the dimension part deviation and the processing parameters.

In literature a number of compensation techniques are described as follows:

- a. by measuring each part immediately after having been machined, the difference between the target surface and the one achieved is determined; the constant component of the difference measured on to the last parts machined is compensated for at the next part;
- b. modeling and compensation of some of the error components (e.g. thermal error) is performed [4, 5];
- c. processing parameters are kept constant in order to obtain uniform deviation which can be further compensated for by the tool offset.

All the papers presented identify and compensate the error components, not the error as a whole [1, 3, 7, 8].

The purpose of the present paper is to develop a new method of online identification of the dimensional dynamics in order to be applied in the dimensional control of a Reconfigurable Machine Tool (RMT). This dimensional control is designed to be adaptive and predictive. The proposed method is experimentally tested in order to evaluate the full-scale performance. The paper has the following structure: chapter 7.2 describes the identification algorithm of the dimensional dynamics; chapter 7.3 shows the experimental research of the dimensional dynamics; finally, chapter 7.4 covers relevant discussions and conclusions.

7.2 Algorithm for online identification of the dimensional dynamics

Let us consider that a RMT has been configured and the machining process is so changed as to be able to process another part of the some products family. As a consequence of these changes the RMT behaviour has been altered therefore the model describing this behaviour is no longer valid [6]. For this reason it is necessary for the RMT to be reidentified. When RMT is machining the part number n of the batch, it is first necessary to identify the relationship between dimensional deviation occurred during part-machining and the RMT monitoring data found during machining of all the previous $n-1$ parts.

In order to describe the stationary state of the system corresponding to a certain bin of the tool trajectory the following variables will be used:

- variables describing the mechanical field; these may represent components of the cutting force or the force the RMT is demanded in its different points ; $F^{(1)}$, $F^{(2)}$, $F^{(3)}$ (its average value in a bin width);
- variables describing the thermal field; these may be average values $T^{(1)}$, $T^{(2)}$, $T^{(3)}$ of the temperature in different points of the RMT;
- variable i represents the count number of the current bin on the tool trajectory;
- variable j represents the count number of the parts group being processed whose RTM behavior and tool wear corresponding to a certain bin remain unchanged. Numbering the parts forming the group has its origin in the current part, (e.g. if, during processing the parts n and $n-1$ the RMT behavior and tool wear have not changed , then n , $n-1$ parts represent the group numbered $k=1$, while parts $n-2$, $n-3$ form a second group for $k=2$ and so on);
- deviation δ corresponding to the bin of count number i .

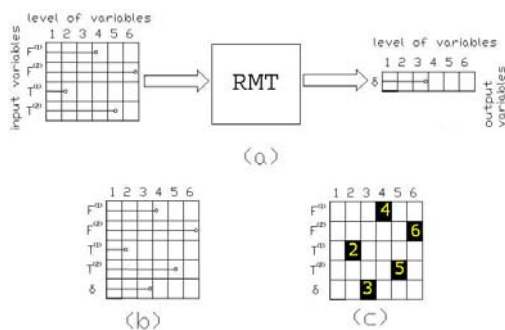
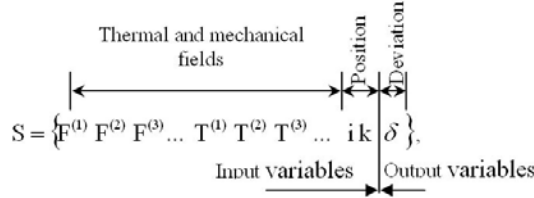


Fig. 7.1. Three representations of the steady state of the system corresponding on the one tool path bin

The values of variables F , T and δ will be scaled on their own variation in such way that all the variables are represented by integer values. For instance in Figure 7.1 these variables have been scaled using six levels.

We define RMT state using an interger number set representing termal and mechanical fields, position and deviation.

The RMT state in a certain moment will be represented as follows:



where $F^{(1)}$, $F^{(2)}$, $F^{(3)}$ stand for the values reached by processing the signals received from the sensors in different points on the system; $T^{(1)}$, $T^{(2)}$, $T^{(3)}$ represent the temperatures measured in different points on the system, i is the current bin count number, and k is the current number of the parts group processed with RMT behavior and tool wear considered unchanged.

For the sake of simplicity let us consider that the mechanical field is evaluated in one single point on the machine. In this case four states can be considered

$$S_1 = \{F_1 T_1 i_1 k_1 \delta_1\}, S_2 = \{F_2 T_2 i_2 k_2 \delta_2\}, \\ S_3 = \{F_3 T_3 i_3 k_3 \delta_3\}, S_4 = \{F_4 T_4 i_4 k_4 \delta_4\}.$$

In the states field the following metrixes can be defined: the difference between S_1 si S_2 states is

$$S_1 - S_2 = \{(F_1 - F_2), (T_1 - T_2), (i_1 - i_2), (k_1 - k_2), (\delta_1 - \delta_2)\} \quad (7.1)$$

and the distance between these states is:

$$d(S_1, S_2) = |F_1 - F_2| + |T_1 - T_2| + |i_1 - i_2| + |k_1 - k_2| \quad (7.2)$$

Several states are considered to belong to the same equivalence class if the input variables meet a certain condition. For instance, we consider states clasification depending on their distance from/to a given state. States at distance a from the given state form equivalence class of a order. Particularly, if $a=0$, then the states are identical.

Transition is defined as a pair of states: an initial state and a final one. Two such transitions are represented below:

$$\tau_{(1,2)} = \left\{ \begin{array}{ccccc} F_1 & T_1 & i_1 & k_1 & \delta_1 \\ \hline F_2 & T_2 & i_2 & k_2 & \delta_2 \end{array} \right\} \begin{array}{l} \downarrow \text{Initial state} \\ \uparrow \text{Final state} \end{array} \quad (7.3)$$

$$\tau_{(3,4)} = \begin{Bmatrix} F_3 & T_3 & i_3 & k_3 & \delta_3 \\ F_4 & T_4 & i_4 & k_4 & \delta_4 \end{Bmatrix}$$

For given transitions we define the difference between two transitions:

$$\tau_{(1,2)} - \tau_{(3,4)} = \begin{Bmatrix} F_1 - F_3 & T_1 - T_3 & i_1 - i_3 & k_1 - k_3 & \delta_1 - \delta_3 \\ F_2 - F_4 & T_2 - T_4 & i_2 - i_4 & k_2 - k_4 & \delta_2 - \delta_4 \end{Bmatrix} \quad (7.4)$$

We consider that two transitions are equivalent if the difference is:

$$\tau_{(p,q)} - \tau_{(r,s)} = \begin{Bmatrix} 0 & 0 & a & b & \delta_p - \delta_r \\ 0 & 0 & a & b & \delta_q - \delta_s \end{Bmatrix} \quad (7.5)$$

where

$$\begin{aligned} a &= i_p - i_r = i_q - i_s \\ b &= k_p - k_r = k_q - k_s \end{aligned}$$

Two equivalent transitions have the same input values variation (mechanical field (**F**), thermal field (**T**) and position (**i**, **k**)). As a result the output values variation will be the same as well:

$$(\delta_p - \delta_r) = (\delta_q - \delta_s). \quad (7.6)$$

Identification algorithm is illustrated in Figure 7.2.

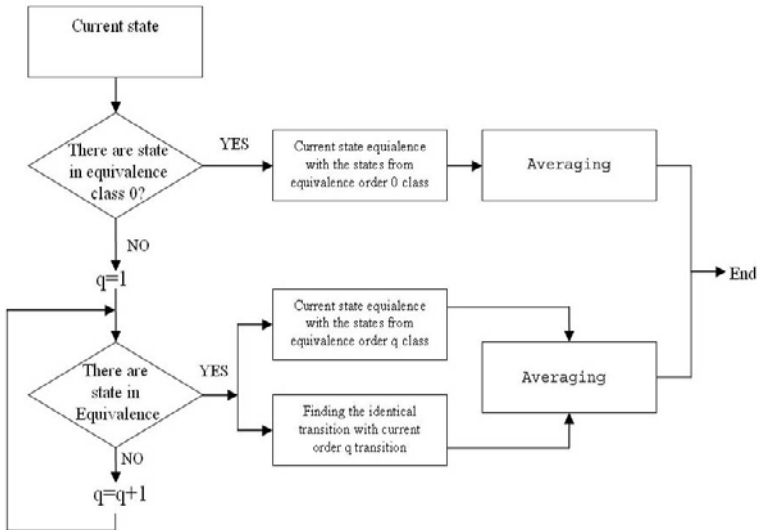


Fig. 7.2. Data processing algorithm

7.3 Experimental research

A number of 70 parts batch has been processed (Figure 7.3). The linear trajectory of the tool was divided into five zones. Dimensional control algorithm was applied separately in each zone. During processing the cutting force was measured. Average value of cutting force was used for the dimensional control of one certain zone on the tool path.

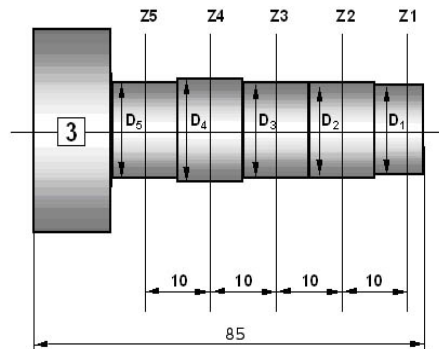


Fig. 7.3. Blank work piece

Using the proposed algorithm, we simulated dimensional compensation and obtained the following results:

- the average value of deviation decreased from 0.837014 mm to 0.0103 mm, meaning an 81 times improvement;
- standard deviation decreased from 0.249 mm to 0.23 mm, that is a 1.1 time improvement.

It is noted that the method performance is good due to the dimensional dynamics of the system being described accurately enough.

Using this method is quite simple, involves a small number of operations to complete prediction; as a result the time taken for prediction calculus is very low. Dimension compensation is performed online which further implies lower costs.

The accuracy can be improved by increasing the number of zones on the tool path, as well as by decreasing the interval width for scaling the process variable values.

7.4 Conclusions

By implementing the proposed method the time needed for RMT identification is significantly reduced. The model resulted from identification is

continuously changing because it takes into account the evolution in time of the RMT and therefore it is able to describe more accurately the real behavior of the system at a certain mom

We believe that the method advanced in this paper is able to improve the RMT behavior during parts machining process, and also to improve the machining process accuracy by error compensation.

The novelty of the proposed method is that, by considering the whole error as an entity, it successfully achieves to compensate the entire deviation occurred during the process.

Acknowledgement. This paper was developed under the research programs CEEEX, grants 22 and 23, funded by the Romanian Ministry of Education and Research.

References

1. Choudhury SK, Jain VK, Rama Krishna S (2001) On-line monitoring of tool wear and control of dimensional inaccuracy in turning. *Journal of Manufacturing Science and Engineering* 123(1):10-12
2. Dong C, Zhang C, Wang B, Zhang G (2002) Prediction and compensation of dynamic errors for coordinate measuring machines. *Journal of Manufacturing Science and Engineering* 124(3):509-514
3. Hong Yang, Jun Ni (2003) Dynamic modeling for machine tool thermal error compensation. *Journal of Manufacturing Science and Engineering* 125(2):245-254
4. Kim S, Landers RG, Ulsoy A (2003) Robust machining force control with process compensation. *Journal of Manufacturing Science and Engineering* 125(3):423-430
5. Lian RJ, Lin BF, Huang JH (2005) A grey prediction fuzzy controller for constant cutting force in turning. *International Journal of Machine Tools and Manufacture* 45(9):1047-1056
6. Mehrabi MG, Ulsoy AG, Koren Y, Heytler P (2000) Trends and perspectives in flexible and reconfigurable manufacturing systems. *Journal of Intelligent Manufacturing* 13(2):135-146
7. Liang Steven Y, Hecker Rogelio L, Landers Robert G (2004) Machining process monitoring and control: The state-of-the-art. *Journal of Manufacturing Science and Engineering* 126(2):297-310
8. Tseng PC, Ho JL (2002) A study of high-precision CNC lathe thermal errors and compensation. *The International Journal of Advanced Manufacturing Technology*, Springer London, 19(11):850-858

Chapter 8

Compaction-induced deformation on flexible substrate

B. Punantapong

Biomaterials and Nanotechnology Research Unit,
Department of Industrial Physics and Medical Instrumentation,
King Mongkut's Institute of Technology North Bangkok
1518 Piboolsongkram Road, Bangsue, Bangkok 10800, Thailand

Abstract. This paper describes the analytical method for evaluation of compaction-induced stresses and deformation on thin layer of flexible substrate by using finite element analysis. The incremental placement and compaction of thin layer of flexible substrate are based on a hysteretic model for residual stresses induced by multiple cycles of loading and unloading. The results showed that the large compaction load can be applied to thin layer of flexible substrate and it achieves higher density effectively. The reinforcement of layer also increases compaction efficiency, because it reduces the ratio between shear and vertical forces during compaction process. The maximum vertical stress on the base of specimen usually decreased with higher compaction thickness. The reinforcement will be increased substrate stiffness under the compaction indenter and it initiates stress concentration. As a result, it maintains higher vertical stress level on the base of specimen that provides better compaction characteristics. Therefore, it can be concluded that the reinforcement is essential to achieve effective compaction on thin layer of flexible substrate.

Keywords. Deformation, Flexible substrate, Hard tissue repair

8.1 Introduction

The design of medical devices is a complex task and its involved the resolution of conflicts and compromise among the desired feature. One of the more inclusive is any substance or combination of substances synthetic or natural in origin, which can be used for any period of time, as a whole or part of a system which treats, augments or replace tissue, organ or function of the body. This definition must be extended because biomaterials are currently being utilized as scaffolds for tissue-engineered devices (hybrids of synthetic or biologic scaffolds and living cells and tissue for vessels, heart valves, and myocardium).

For example, the cardiovascular biomaterials may contact blood (both arterial and venous), vascular endothelial cells, fibroblasts, as well as a number of other cells and a cellular matrix material that make up all biological tissue. At the same time, orthopedic biomaterials were the complex and important role of the cellular component of bone. Then the general properties of bone, ligament, tendon, and cartilage have been well characterized over the past century. So, orthopedic tissues are affected by both the stresses that they experience on a dialy basis and as a result of trauma and disease processes [1]. Many of these injuries or pathologies require medical intervention that may be assisted through the use of engineered materials.

Thus biomaterials used in clinical system are susceptible to a number of failure modes. Like all materials, mechanical failure is possible, particularly in implants (expected to exceed 10 years). However, advances in biomaterials have been focused on the design of dental and orthopedic implants, which are mainly structural applications. So, tremendous stiffness and strength improvement is not always the concern in the design of biomaterials and even less so for large weight savings. Other concerns, such as biocompatibility, precise property matching, mimicking natural structures, etc., can become more important, yet these too are areas where biomaterials offer much promise in device design.

Furthermore, It is very difficult to achieve good strength and stiffness for flexible substrate of biomaterials. Hence, flexible substrate is not used for reinforced substrate structures, including solid-reinforced substrate. However, applying tensile reinforcement or compression with preloading and prestressing procedures only allows minimal deformation in the body of compacted substrate. Recently, de Boer et al [2] proposed an engineering model that captures key grain scale phenomena in a manner that is consistent with both the experimentally measured bulk material behavior and grain contact mechanics. The model predictions agree well with exper-

rimental results for the explosion threshold of granular due to weak mechanical impact. Because the work was restricted to weak impact scenarios for which the average intergranular stress was much less than the bulk modulus of the pure phase solid, an incompressible solid was assumed. While this approach can provide meaningful statistical information about hot spot fluctuations, it requires extensive computational resources and time, and thus cannot be applied to engineering scale systems that contain in excess of billion individual grains. This approach has proven to be successful in understanding the bulk granular material response induced by various impact conditions [3].

The present study investigates the effectiveness of a composite reinforcement on thin layer of flexible substrate during compaction. Other simulation concepts like the finite element method mostly are used to investigate the phenomena of substrate layer interaction under quasi-static condition like constant loading. So, it is necessary to take into account the important influence of the layer on the dynamic behavior of substrate. Furthermore, the dynamical layer-load-changes enlarge, loading to an increased deformation of the surface. The following axle of the substrate will be excited by the back unevenness of the first axle, which again induces the excitation of the substrate.

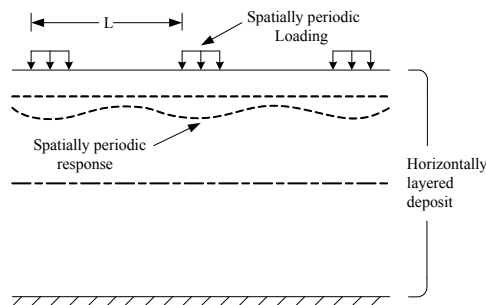


Fig. 8.1. Spatially periodic loading on a horizontally layered deposit (adapted from [5])

8.2 Materials and method

The finite element analysis methodology developed for analysis of stresses and deformations. The resulting from placement and compaction of layers of fill simulates the actual sequence of field operations in a number of sequential steps. These analytical procedures permit modeling of multiple cycles of compaction loading at any given fill stage with a single solution

increment. In these analyses, compaction loading is realistically considered as a transient moving concentrated surficial load which may pass more times over some specified portion of the surface. In Figure 8.1 shows a spatially periodic loading (of period L) is applied to the surface of a horizontally layered profile of an elastic layer.

Therefore, a strip loading as a spatially periodic loading is applied to the surface of a horizontally layered profile of a flexible substrate. It is well known for such a periodic loading (or loading function). A Fourier series representation can be used Nemat-Nasser [4]. For instance, if the Cartesian coordinate in the horizontal direction is x and the loading function is $p(x)$ then it is found that

$$p(x) = \sum_{n=0}^{\infty} P^{(n)} \cos \alpha_n x \quad (8.1)$$

where

$$P^n = \frac{2}{L} \int_0^L p(x) \cos \alpha_n x dx \quad (n > 0)$$

$$P^n = \frac{1}{L} \int_0^L p(x) dx \quad (n = 0)$$

and $\alpha_n = 2n\pi / L$

The loading has been represented by the sum of periodic functions (in this case cosine functions because the loading function $p(x)$ was chosen to be an even function of x). It may be observed that, for such a spatially periodic loading, the displacements in the substrate below are also periodic. It is therefore possible to write the displacements as a series of periodic functions [5].

$$u_x = \sum_{n=0}^{\infty} U^{(n)} \sin \alpha_n x \quad (8.2a)$$

$$u_z = \sum_{n=0}^{\infty} W^{(n)} \cos \alpha_n x \quad (8.2b)$$

It will be observed that whereas a series of cosine functions is used to represent the displacement in the z direction (u_z), a sine series is used for the lateral displacements in the lateral or x direction (u_x). This follows from the observation that the lateral displacements are anti-symmetric about the axis of symmetry of each loaded area. The coefficients in the above series

$U^{(n)}$, $W^{(n)}$ are of course different at different depths z since both the displacement components vary throughout the depth of the layer.

$$u_x(x, z) = \sum_{n=0}^{\infty} U^{(n)}(z) \sin \alpha_n x \quad (8.3a)$$

$$u_z(x, z) = \sum_{n=0}^{\infty} W^{(n)}(z) \cos \alpha_n x \quad (8.3b)$$

However, the shorthand notation of equations (8.2a and 8.2b) will be used in situations in which no confusion arises. Suppose that we now take one term of the cosine series representing the loading function $p(x)$, and obtain the solution to the problem associated with this single sinusoidal load applied to the surface of the substrate layer.

$$p(x) = P_n \cos \alpha_n x$$

Then

$$u_x = U_n(z) \sin \alpha_n x \quad (8.4a)$$

$$u_z = W_n(z) \cos \alpha_n x \quad (8.4b)$$

At the same time, we will be assumed that conditions of plane strain prevail so that there is no displacement in the y direction and no variation of the field quantities with y . Then it is only necessary to consider stress and displacement fields having the form

$$u_x = U(\alpha, z) \sin \alpha x \quad (8.5a)$$

$$u_z = W(\alpha, z) \cos \alpha x \quad (8.5b)$$

$$\sigma_{xz} = T(\alpha, z) \sin \alpha x \quad (8.5c)$$

$$\sigma_{zz} = N(\alpha, z) \cos \alpha x \quad (8.5d)$$

$$\sigma_{xx} = H(\alpha, z) \cos \alpha x \quad (8.5e)$$

$$\sigma_{yy} = M(\alpha, z) \cos \alpha x \quad (8.5f)$$

since the solution for more complex load cases can be obtained by superposition of components given by equations (8.5) and where it has been assumed that α stands for any particular value of $\alpha_n = 2n\pi/L$ and $U^{(n)} = U(\alpha_n, z)$, etc.

Let us now consider the determination of the field quantities in a particular layer, l of the material [6]. If equations (8.5) are substituted into the equilibrium equations it is found that

$$-\alpha H + \frac{\partial T}{\partial z} = 0 \quad (8.6a)$$

$$-\alpha T + \frac{\partial N}{\partial z} = 0 \quad (8.6b)$$

Similarly if equations (8.5a-8.5f) are substituted into Hooke's Law it is found that

$$\alpha U = \frac{1}{E_l} [H - \nu_l (M + N)] \quad (8.7a)$$

$$\frac{\partial W}{\partial z} = \frac{1}{E_l} [N - \nu_l (H + M)] \quad (8.7b)$$

$$\frac{\partial U}{\partial z} - \alpha W = \frac{2(1 + \nu_l)T}{E_l} \quad (8.7c)$$

$$M = \nu_l (H + N) \quad (8.7d)$$

where E_l, ν_l denote the values of Young's Modulus and Poisson's ratio for layer, l .

Equations (8.6), (8.7) allow the non-zero stress and displacement to be expressed in terms of N and thus

$$H = -\frac{\partial^2 N}{\partial Z^2} \quad \text{and} \quad T = -\frac{\partial N}{\partial Z} \quad (8.8)$$

$$M = \nu_l \left(N - \frac{\partial^2 N}{\partial Z^2} \right)$$

$$\alpha E^* U = -\frac{\partial^2 N}{\partial Z^2} - \nu^* N$$

$$\alpha E^* W = -\frac{\partial^3 N}{\partial Z^3} + (2 + \nu^*) N$$

where $E^* = \frac{E_l}{1 - \nu_l^2}$, $\nu^* = \frac{\nu_l}{1 - \nu_l}$ and $Z = \alpha z$

and

$$\frac{\partial^4 N}{\partial Z^4} - 2 \frac{\partial^2 N}{\partial Z^2} + N = 0 \quad (8.9a)$$

If equation (8.9a) is now solved, we obtain

$$N = X_1 C + X_2 Z + X_3 S + X_4 ZC \quad (8.9b)$$

The fundamental step in the finite layer technique is to determine the four constants $X_1, \dots, X_4$ in terms of boundary quantities. To be more precise, suppose that layer l is bounded by the node planes ($z = z_l$) and ($z = z_m$) where $m = l + 1$ and that the subscripts l, m indicate the value of a particular quantity on the indicated node plane. Then the solution can be used to determine $X_1, \dots, X_4$ in terms of U_l, W_l, U_m, W_m . Once $X_1, \dots, X_4$ are known they can be used to evaluate T_b, N_b, T_m, N_m and so establish a relationship of the form

$$\begin{bmatrix} -T_l \\ -N_l \\ +T_m \\ +N_m \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \\ k_{41} & k_{42} & k_{43} & k_{44} \end{bmatrix} \begin{bmatrix} U_l \\ W_l \\ U_m \\ W_m \end{bmatrix} \quad (8.10)$$

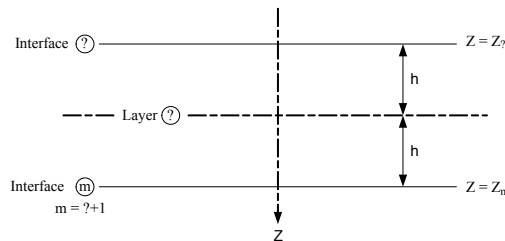


Fig. 8.2. Single cell layer within the surface of flexible substrate

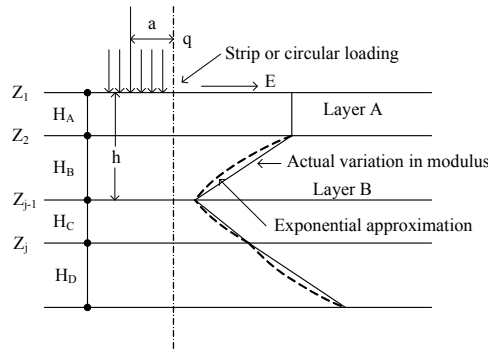


Fig. 8.3. Non-homogeneous profile tested with exponential approximation (adapted from [6])

The matrix occurring in equation (8.10) is called the layer stiffness matrix of the particular layer (layer, l in this case). Layer stiffness matrices can be used to construct solutions for layered deposits in exactly the same way as element stiffness matrices are used in conventional applications of the finite element method.

Consequently, we present a novel way to study the behavior of layer of flexible material when it received forces by compression. Thus FEM technique is necessary to use a dynamic solving algorithm where not only the displacements of the nodes, but as coming from the Newton's dynamic equilibrium the accelerations $\ddot{\mathbf{u}}$ for each node in the model are considered as Hiroma et al. [7]

$$\ddot{\mathbf{u}}_{(i)} = \mathbf{M}^{-1} (\mathbf{P}_{(i)} - \mathbf{I}_{(i)}) \quad (8.11)$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{P}$ is the vector of the external forces (i.e. contact forces or nodal forces of neighboring elements) and $\mathbf{I}$ is the vector of internal forces on the basis of the stress inside the elements.

The mass matrix can be obtained by distributing the mass of each element on to its nodes. The created point mass belonging to a node is the sum of all partial masses of all the elements defined at this node. Using the explicit solving algorithm, available with ANSYS, the velocities $\dot{\mathbf{u}}$ and the displacement $\mathbf{u}$ for all degrees of freedom can be calculated directly from the information at the beginning of the time increment. The velocities and the displacements were integrated over the time according the following formular.

$$\dot{\mathbf{u}}_{(i+1/2)} = \dot{\mathbf{u}}_{(i-1/2)} + \frac{\Delta t_{(i+1)} + \Delta t_{(i)}}{2} \cdot \ddot{\mathbf{u}}_{(i)} \quad (8.12)$$

$$\mathbf{u}_{(i+1)} = \mathbf{u}_{(i)} + \Delta t_{(i+1)} \dot{\mathbf{u}}_{(i+1/2)} \quad (8.13)$$

The material property of the flexible substrate in this simulation corresponds to granule with the following parameters: cohesion, $c = 0.029$ N/mm²; angle of internal friction, $\varphi = 22.5^\circ$; density, $\rho = 2.06$ t/mm³; elasticity, $E = 20$ N/mm. The dynamic behavior of the flexible substrate is condensed on the damping by the energy loss of the plastic deformation and on the effect of mass-inertia of substrate elements in this simulation.

8.3 Results and discussion

Results from such an analysis are shown in Figures 8.4 and 8.5. A circular and a rectangular loading have been considered. Each loading is applied to the surface of a substrate layer which is made up of two sublayers; sublayer (A) is of depth H_A while the lower layer (B) is of depth H_B . The loadings were chosen so that they had the same minimum dimension, i.e. the loading, q was applied over the region $|x| < a$ (strip), $0 < r \leq a$ (circle), $|x| < a$, $|y| < b$ (rectangular). The material was assumed to be anisotropic. For layer A, $E_h / E_v = 1.5$, $G / E_v = 0.45$, $\nu_h = 0.25$, $\nu_{hv} = 0.3$, $\nu_{vh} = 0.2$ and for layer B, $E_h / E_v = 33$, $G / E_v = 0.5$, $\nu_h = 0.1$, $\nu_{hv} = 0.9$, $\nu_{vh} = 0.3$ (The subscripts h , v indicate horizontal and vertical directions, respectively.).

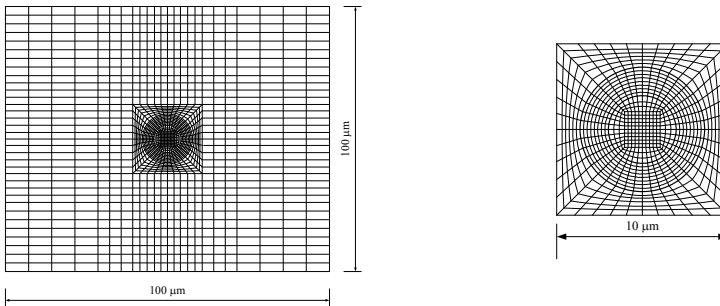


Fig. 8.4. Surface pressure mobilization of single cell on the surface of flexible substrate

The plots shown in Figure 8.5 is for the vertical σ_{zz} stresses along the axis of the loading ($x = y = r = 0$). There is a large difference in the vertical stress computed for each of the loading types. However, there is less difference in the horizontal stresses for this particular substrate profile.

All of the problems discussed thus far have been concerned with vertical loading, however there are many engineering problems which involve lateral or horizontal loadings applied to foundations. An example of study is

shown the result in Figure 8.6, where a uniform shear, it is shown applied to the surface of a thin layered flexible substrate over a circular region of radius, a . The variation of vertical, u_z and horizontal u_x displacements with depth is calculated for the three layered system as shown in Figure 8.6 ($x/a = 0.5, y/a = 0$). For this experiment the material properties of each of the sublayers A, B, C are described below:

- Layer A: $E_h / E_v = 2, G_v / E_v = 0.4, \nu_h = 0.3, \nu_{vh} = 0.2$
- Layer B: $E_h / E_v = 2, G_v / E_v = 0.4, \nu_h = 0.3, \nu_{vh} = 0.2$
- Layer C: $E_h / E_v = 1, G_v / E_v = 1/3, \nu_h = 0.5, \nu_{vh} = 0.5$

and the ratio of Young's modulus in the layers is assumed to be

$$(E_v)A : (E_v)B : (E_v)C = 25 : 5 : 1$$

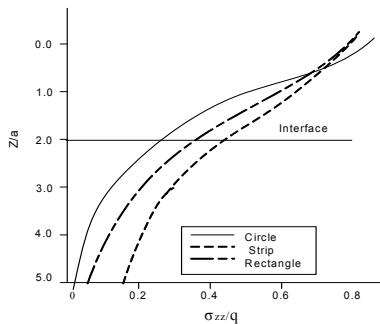


Fig. 8.5. Stresses in a substrate consisting of two anisotropic layers.

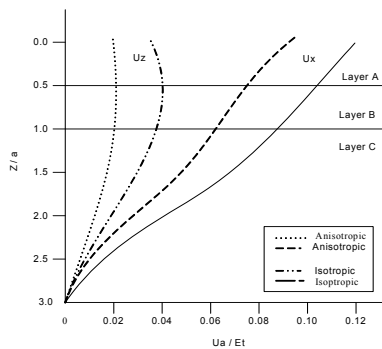


Fig. 8.6. Horizontal and vertical displacements in layered material

Figure 8.7 shows the time-settlement behavior of strip loading on a substrate layer with a shear modulus which increases with depth (non-dimensional time, τ_0 where $\tau_0 = c_0 t / l^2$). The coefficient of consideration, c_0 is defined on the figure in terms of the Poisson's ratio of the substrate, ν

and its permeability per unit weight of water k / γ_w . It may be observed from the time settlement plots that as the rate of increase of shear modulus with depth l_m / l becomes larger the final deflections become smaller and occur at smaller time, τ_0 .

Problems involving the time-dependent deformation of materials under a constant applied loading may often be treated by assuming that the materials display viscoelastic behavior, i.e. that the material properties are themselves dependent on time. In this study, we used the assumption that the behavior of the material may be separated into a deviatoric and a volumetric behavior. That is to say there will be a different time-dependent response to a mean stress increase, then to a deviatoric or shear stress increased.

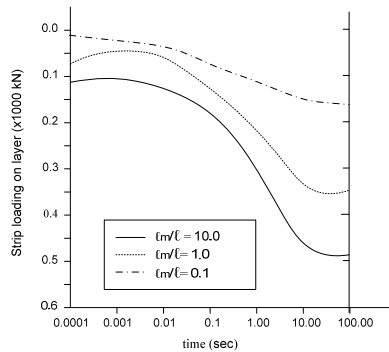


Fig. 8.7. Time-settlement behavior of strip loading on a substrate layer with a shear modulus

In Figure 8.8 shows an example of the relation of deflection vs time for central point of a strip or circular loading resting on a layered material. It involves loading applied to a flexible substrate which consists of an upper viscoelastic layer and a lower elastic layer. Both materials were assumed to have a constant Poisson's ration, ν , that of the upper layer being $\nu = 0.5$ (i.e. incompressible) and that of the lower layer $\nu = 0.3$. It was assumed that the deformation of the material was due to a time-dependent shear modulus.

8.4 Conclusions

In the paper a two dimensional FEM-model of the compaction-induced deformation on thin layer of flexible substrate was presented. With this extension to the simulation concept, the new concept based on the dynamical FE-simulation was developed, that includes the possibility to investigate

the dynamical load changes, caused by vertical oscillation of indenter and the according substrate deformation. It was pointed out, that the indenter oscillation takes influence to the mobility of the substrate. Due to these oscillations, the dynamical compaction on thin layer of substrate changes enlarge, leading to an increased deformation of the surface, which again induces the excitation of the substrate. Therefore, dynamic loads and substrate deformation influence mutually.

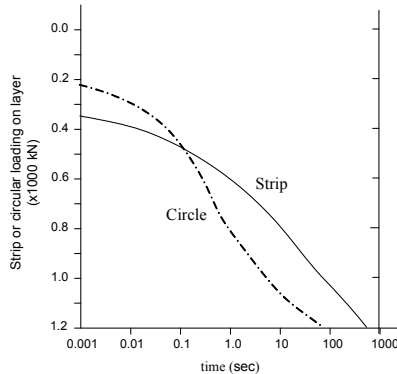


Fig. 8.8. Deflection vs time for central point of a strip or circular loading resting on a layer material with a viscoelastic upper layer

References

1. Kaasschieter EE, Frijns A-JH, Huyghe JM (2003) Mixed finite element modeling of cartilaginous tissues. *Mathematics and Computers in Simulations* 61:549-560
2. de Boer R, Didwania AK (2004) Two-phase flow and the capillarity phenomenon in porous solids. *Transport Porous Med* 56:137-170
3. Philippe P, Bideau D (2002) Compaction dynamics of granular medium under vertical tapping. *Erophys. Letter* 60:677
4. Nemat-Nasser S (1982) On finite deformation elasto-plasticity. *Journal of Solids Structure* 18:857-872
5. Small JC, Brown PT (1988) Finite layer analysis of the effects of a sub-surface load. In: *Proc. 5th Aust-NZ. Conf. Geomechanics*, Sydney, pp 123-127
6. Small JC, Booker JR (1986) Finite layer analysis of layered elastic materials using a flexibility approach Part 2. *Journal of Methods Engineering* 23:959-978
7. Hiroma T, Wanjii S, Kataoka T, Ota Y (1997) Stress analysis using FEM on stress distribution under a wheel considering friction with adhesion between a wheel and soil. *Journal of Terramechanics* 34(4):35-42

Chapter 9

Robust and accurate visual odometry by stereo

Aldo Cumani, Antonio Guiducci

Istituto Nazionale di Ricerca Metrologica,
Str. delle Cacce, 91, I-10135 Torino, Italy
{cumani,guiducci}@inrim.it

Abstract. This work deals with the problem of estimating the trajectory of an autonomous rover by onboard passive stereo vision only (so-called *Visual Odometry*). The proposed method relies on bundle adjustment of tracked SURF feature points. Tests on real-world data show the effectiveness of this approach. In the case of cyclic trajectories, even greater accuracy is attained by periodically comparing the scene against a set of stored references.

Keywords. Robot localisation, Egomotion, Stereo vision, Bundle adjustment, SLAM, SURF

9.1 Introduction

In the framework of research on Simultaneous Localisation and Mapping (SLAM), our group at INRIM has developed a visual odometry algorithm [1] which relies on the tracking of pointwise features extracted from the images acquired by an onboard binocular stereo head. At more or less regular intervals along the rover trajectory, its motion is estimated by robust bundle adjustment of the tracked features in the four images (two before and two after the motion).

Extensive tests have been performed using Shi-Tomasi-Kanade features [2], which however require small image displacements for a reliable tracking, and a stereo disparity cue (from a correlation disparity map) for stereo

matching. We have therefore switched to SURF features [3], which appear to be more robust, allowing direct stereo matching and tracking with longer inter-frame steps. Moreover, the SURF features collected from an image appear to be a representation of the image contents sufficiently good for subsequent image-based matching.

Indeed, any localisation algorithm based on the integration of subsequent motion estimates is affected by error accumulation. In the case of cyclic trajectories, i.e. with the rover returning to some previously visited place, it is important that the algorithm be able to recognize the latter situation and then perform some adjustment to its estimated pose.

This can be achieved by periodically storing the observed features and comparing the current features to the previously stored ones; when there is a sufficient number of matches, the bundle adjustment algorithm can then yield an estimate of the current rover pose, relative to the stored one, which is expected to be more accurate than the one obtained by the integration of the preceding steps along the path. This *déjà vu* feature has been incorporated in our algorithm, with rather encouraging results.

The following sections describe the algorithm in some detail, and present some results from actual test runs.

9.2 The visual odometry algorithm

Our algorithm relies on estimating the relative motion from corresponding features in the images acquired while the rover is moving. Such estimates are computed at *key frames*, whose spacing is a compromise between larger intervals, which are desirable for numerical accuracy, and the need for a sufficiently large number of features, which naturally tend to be lost either because going out of view or because of matching errors. The accumulation of errors is overcome, in the case the rover returns to some previously visited place, by recognizing that this situation occurs and correcting the estimated rover pose.

The algorithm can be summarized in the following steps:

- **Feature extraction.** Point features are extracted at each key frame and left-right matched.
- **Feature tracking.** Matched features are tracked in subsequent frames, where stereo matching is performed again.
- **Motion estimation.** The relative motion of the rover between the current frame and the last key frame is estimated by bundle adjustment of matched feature points.

- **Key frame declaration.** When the performance of the algorithm decays, due e.g. to the presence of too few matched points close to the rover, or to a failure of the bundle adjustment step either for numerical errors or because of an excessive number of outliers, a key frame is declared on the previous frame.
- **Cyclic correction.** Observed features and corresponding pose estimates are periodically saved to an archive. When the rover believes to be near some saved position (and with a similar heading), the detected features are compared to the stored ones. If enough matches are found, a bundle adjustment step between the current features and the stored ones yields the pose correction.

9.2.1 Feature extraction and tracking

The so-called *Speeded-Up Robust Features* [3] describe a distribution of Haar-wavelet responses within the neighbourhood of some interest points, the latter having been determined by a Hessian-based detector.

Being largely insensitive to scale and rotation, as well as to illumination, SURFs have proven quite robust and effective, both for what concerns stereo (left-right) matching, as well as for tracking along the sequence. While being in some way similar to the SIFT features used by other researchers [4], SURFs have over the latter the distinct advantage of being unencumbered - at least for now - by patent issues.

Last but not least, in our experience the SURF features collected from an image appear to be a representation of the image contents sufficiently good for subsequent image matching, as explained in Sec. 9.2.3.

In order to get even stronger insensitivity to lighting conditions, in our implementation SURFs are not computed directly from the graylevel images, but from LOG filtered versions of the same. Both left-right and temporal matching relies on a feature similarity measure defined in terms of the Euclidean distance between feature descriptors. The matching algorithm can be improved by exploiting specific matching constraints (e.g. positive disparity for the stereo case), though in fact even the simplest approach already yields very good results. In our implementation, only stereo matching constraints are imposed, i.e. positive disparity and vertical distance under some threshold. No constraint is imposed instead on temporal matching.

Stereo matched point pairs are backprojected to a 3D point estimate, in the camera reference. This 3D estimate is used as a starting point for the bundle adjustment step. The features detected in the left image at a key frame are then tracked along the sequence up to the next key frame.

9.2.2 Motion estimation

At every key frame, say $k + 1$, except the first, we have a set of N_k features F_i , left-right matched and tracked from the previous key frame k to the next one $k + 1$ (for the sake of simplicity, and without loss of generality, from now on we shall assume $k = 1$ and drop the k index). Let $\mathbf{X}_i = [x_i, y_i, z_i, t_i]^T$ be the unknown 3D projective coordinates of feature F_i in the reference of the left camera at frame 1. Note that the scale ambiguity on $\mathbf{X}_i$ can be fixed by taking $z_i = 1$ since all the points must be visible in the first left image, hence they have strictly positive depth z . Let $\mathbf{u}_{iq} = [u_{iq}, v_{iq}]^T$ be the 2D image coordinates of the feature point in image $q \in \{1L, 1R, 2L, 2R\}$, and $\mathbf{x}_{iq} = [su_{iq}, sv_{iq}, s]^T$ the same in projective representation. Then

$$\begin{aligned} \mathbf{x}_{i,1L} &= \mathbf{P}_L \mathbf{X}_i \\ \mathbf{x}_{i,1R} &= \mathbf{P}_R \mathbf{M}_S \mathbf{X}_i \\ \mathbf{x}_{i,2L} &= \mathbf{P}_L \mathbf{M}_{12} \mathbf{X}_i \\ \mathbf{x}_{i,2R} &= \mathbf{P}_R \mathbf{M}_S \mathbf{M}_{12} \mathbf{X}_i \end{aligned} \quad (9.1)$$

where $\mathbf{P}_L$ and $\mathbf{P}_R$ are the 3×4 intrinsic camera matrices, and $\mathbf{M}_S$, $\mathbf{M}_{12}$ are 4×4 Euclidean transformation matrices representing the stereo (left-to-right) mapping and the motion (frame 1 to frame 2) mapping, respectively, and are of the form

$$\mathbf{M} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} = \begin{bmatrix} e^{[\mathbf{r}]_\times} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (9.2)$$

where $\mathbf{r}$ is the vector representation of rotation $\mathbf{R}$, i.e. $\mathbf{r}/\|\mathbf{r}\|$ = rotation axis, $\|\mathbf{r}\|$ = rotation angle, and $[\mathbf{r}]_\times$ is the antisymmetric matrix representation of vector cross product by $\mathbf{r}$.

Note that in the above both $\mathbf{P}_L$, $\mathbf{P}_R$ and the stereo transform $\mathbf{M}_S$ are known from calibration data (in fact, in our implementation we pre-rectify the stereo pair so that $\mathbf{P}_S \equiv \mathbf{I}_{3 \times 3}$ and $\mathbf{t}_S = [b, 0, 0]^T$ with b the stereo baseline length). We can also take $\mathbf{P}_{L,R} \equiv [\mathbf{I}_{3 \times 3}, \mathbf{0}_3]$ in Eq. (9.1) if the pixel-based feature coordinates are preliminarily transformed to *calibrated* coordinates using the actual intrinsic camera matrices. Now, for each pair of consecu-

tive key frames we have measurements u_{iq}^* of the quantities u_{iq} corresponding to Eq. (9.1), so we can define an error measure

$$\begin{aligned} J(p) &= \sum_i \sum_q f\left(\|e_{iq}\|^2\right) \\ &= \sum_i \sum_q f\left(\|u_{iq} - u_{iq}^*\|^2\right) \end{aligned} \quad (9.3)$$

parametrized as a function of the $6 + 3N$ unknowns

$$p = [r_{12}, t_{12}, x_1, y_1, t_1, \dots, x_N, y_N, t_N]^T \quad (9.4)$$

with r_{12}, t_{12} the generators of M_{12} . The visual odometry estimate is then given by the r_{12}, t_{12} entries of the p value that minimizes J .

In the above, choosing $f(\cdot) = \text{identity}$ corresponds to standard least squares, which would be optimal for independent Gaussian distributed image errors. For reasons of robustness against outliers (mismatched points), it is however preferable to use a robust cost function such as the Lorentzian cost:

$$f(e^2) = \log(1 + e^2/\sigma^2) \quad (9.5)$$

with σ chosen as a function of the expected image-plane error.

Since the first derivatives of J with respect to p can be computed analytically, the minimization of J can be achieved by a gradient-based method such as Levenberg-Marquardt. Moreover, for given i the corresponding terms in Eq. (9.3) only depend upon one of the triples (x_i, y_i, t_i) , which allows to use a sparse algorithm like the one described in [5, A4.3].

For better accuracy, the residuals e_{iq} are compared against a threshold (proportional to the Lorentz σ), and those exceeding that threshold are marked as outliers. The bundle adjustment is then run again on the inliers only.

9.2.3 Cyclic correction

In order to be able to implement cyclic correction, the rover keeps an archive of scene descriptions. Each such description consists of the set of matched features in the left and right images, together with the estimated rover pose.

The description of the current scene is added to the top of the archive when:

- enough time is elapsed since the last entry in the archive (the top entry);
- the number of matched features between the frame to be added and the top entry is near zero.

This strategy is aimed to ensure that the scenes corresponding to the archive entries are all different from each other.

As soon as a new key frame is processed, the estimated rover pose is checked against those stored in the archive, starting from the bottom (i.e. oldest) entry. If an archive entry with similar pose is found, the features in the current frame and those stored in the archive entry are compared, and if enough matches are found a pose correction between the archived and the actual pose is computed by bundle adjustment. All the corrections computed while the rover remains near an archived position are saved in memory. When the rover leaves the neighborhood of that archived position, the saved correction corresponding to the frame with the highest number of matched features is picked and actually applied. The pose corrected this way is expected to be more accurate, as the error accumulated along the intervening path is exchanged with that from a single keyframe pair.

9.3 Experimental results

9.3.1 Equipment and setup

The proposed method has been implemented in C and MATLAB, except for the SURF computation step, where we have used the authors' implementation available on the Web [3]. The algorithm has been tested on sequences of real images acquired by a small commercial rover, an ActivMedia Pioneer 3-DX, equipped with a suitable stereo head (Figure 9.1).

While previous tests had been done with a Videre Design STH-MDCS stereo head, a low-cost commercial product with a stereo baseline of 88 mm, for the tests reported here we have used a better quality, home-made stereo head. This head consists in a pair of Basler A312f digital cameras, equipped with 6mm lenses and mounted on an aluminum slab, as shown in the figure. Each camera provides a stream of CCIR-size (720×576) 8-bit graylevel images on a IEEE1394 bus, and the two cameras can be accurately synchronized by an external trigger signal.

The cameras were mounted at a nominal center-to-center distance (stereo baseline) of 236 mm, and the head was calibrated using Bouguet's method [6], with a 0.6×0.6 m checkerboard pattern.



Fig. 9.1. Detail of the Pioneer 3DX rover with stereo head

In order to have some ground-truth rover position data against which to compare the outcome of the visual odometry algorithm, the following setup was used. As shown in Figure 9.2, three tennis balls were laid on the test field, approximately at the corners of an equilateral triangle (the fourth central ball visible in the figure is just for check). Their distances were manually measured to within ± 1 mm, so defining a metric reference frame. As shown in the figures, another ball was rigidly attached to the stereo head, to be used as reference point for the rover position. Two compact digital cameras (one with 2048×1536 and one with 2592×1944 resolution), after intrinsic calibration again by Bouguet's algorithm, were placed on tripods and used to frame the scene at about 90° with respect to the center of the reference frame. At regular intervals, the robot was stopped and images of the scene were taken by the two cameras; the ball centers, manually extracted from each image pair, allowed then to estimate a ground-truth position of the rover in the chosen reference, to within an accuracy surely better than the ball diameter (about 0.06m).



Fig. 9.2. The Pioneer 3DX rover on the test field. The tennis balls laid on the ground are used as reference points for the estimation of a ground-truth rover position independent from the visual odometry algorithm

9.3.2 Test results

Some results from a test tour are reported in the following. The rover was driven to perform multiple loops over the field framed by the two still cameras, returning several times, with the same heading, to three selected points (including the origin).

The total path length was about 78m. In Figure 9.3 the lighter curve is the rover path as estimated by the visual algorithm alone, i.e. without the déjà vu corrections, while the darker one takes into account the latter ones.

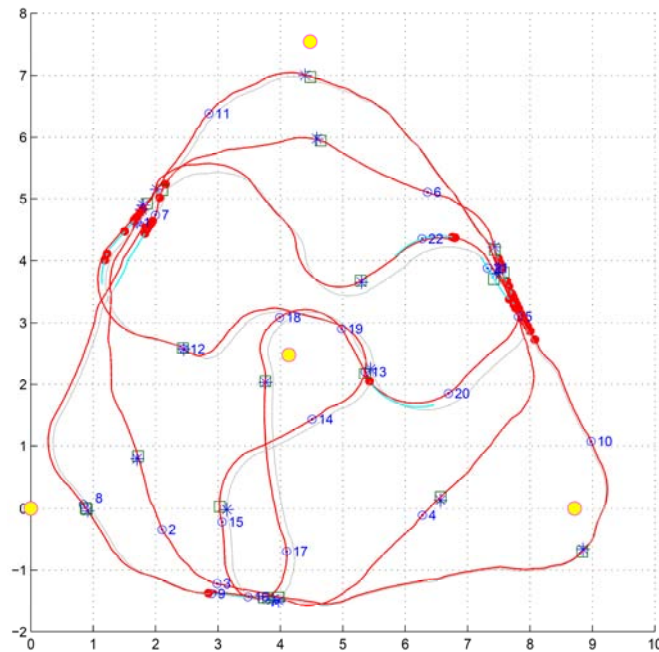


Fig. 9.3. Estimated rover trajectory without (light gray) and with (red/dark gray) corrections. Coordinates in m. Hollow circles indicate saved positions. Dots indicate points where the scene is checked against the archive, bigger ones are those with enough matches to allow computation of the correction

Numbered and circled points are the saved positions. The other dots on the trajectory are the positions where some match against the stored archive is attempted and the number of matches is deemed sufficient for computing the correction.

For comparison with the ground truth, the asterisks mark the estimated rover positions as obtained by triangulation over the images from the two still cameras, while nearby squares are the corresponding estimates from visual odometry.

From this plot the proposed visual odometry method seems to yield quite good results even without cyclic corrections; indeed, the max position error at the measured stops is less than 0.3m over a path length exceeding 75m (i.e. $< 0.4\%$). However, cyclic corrections can still improve the estimated trajectory, as shown in Figure 9.4.

The latter plots the position error, i.e. the distance of the visual odometry position estimate from the point measured via triangulation, as a function of path length. As shown in the figure, while the uncorrected position error grows with path length, the application of the cyclic corrections reduces the maximum error to about half ($\sim 0.15\text{m}$) the value for the uncorrected case ($\sim 0.3\text{m}$). In evaluating these figures, one should also take into account that they are not much greater than the uncertainty of the ground truth values, which with our setup is of the order of several 0.01m.

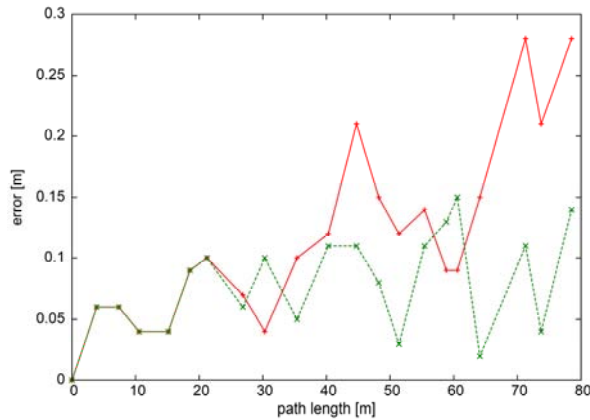


Fig. 9.4. Position error as a function of path length, without corrections (solid/red curve) and with cyclic corrections (green/dashed curve)

9.4 Concluding remarks

The preceding discussion has shown that onboard passive stereo vision alone is able to yield an accurate estimate of the trajectory of the rover, even on an almost featureless ground like the one used in our tests. The error figures here reported are well under e.g. those reported by the Mars Rover team [7]. We have also presented a mechanism that, in case of cyclic trajectories, is able to automatically compensate the accumulated error.

It should be noted that our approach has some similarity with the one reported in [8]. Those authors however, have a quite different approach

(RANSAC) to the problem of outliers (wrong matches). Given that the feature detection/matching scheme adopted in our work typically yields rather few outliers, our approach, using a robust cost function, seems both effective and less computationally expensive.

There is however still room for improvements; for example, the current implementation finds corresponding image points only by comparing independently extracted interest points. Adding e.g. a step to refine feature position by local correlation (as with the Shi-Tomasi-Kanade features) could possibly yield even more accurate results. More extensive tests, especially on longer paths, are also needed to assess the practical usability of this method for unassisted navigation.

Acknowledgement. The authors wish to thank their colleague Giorgio Quaglia for his invaluable help in setting up the stereo head and the related acquisition software.

References

1. Cumani A, Guiducci A (2006) Stereo-based visual odometry for robust rover navigation. WSEAS Trans. Circuits and Systems 5(10):1556–1562
2. Shi J, Tomasi C (1994) Good features to track. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp 593–600
3. Bay H, Tuytelaars T, Van Gool L (2006) SURF: Speeded up robust features. In: Proceedings 9th European Conference on Computer Vision, See also <http://www.vision.ee.ethz.ch/~surf/>
4. Se S, Lowe D, Little J (2001) Vision-based mobile robot localization and mapping using scale-invariant features. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp 2051–2058
5. Hartley R, Zisserman A (2003) Multiple View Geometry in Computer Vision. Cambridge Univ. Press, 2nd ed.
6. Bouguet JY (2004) Complete camera calibration toolbox for MATLAB. Technical report, <http://www.vision.caltech.edu/bouguetj/calibdoc/index.html>
7. Cheng Y, Maimone M, Matthies L (2005) Visual odometry on the Mars exploration rovers. In: Proc. IEEE Int. Conf. Systems, Man and Cybernetics, 1:903–910
8. Sünderhauf N, Protzel P (2006) Towards using sparse bundle adjustment for robust stereo odometry in outdoor terrain. In: Proc. Int. Conf. TAROS 2006, pp 206–213

Chapter 10

Financial auditing in e-business environment

L. D. Genete, Al. Țugui

Business Information System Department, “Alexandru Ioan Cuza” University of Iasi, 22 Carol I Blvd., Ro-700505, ROMANIA

Abstract. This study has the purpose to specify the main changes determined by e-business in the field of financial audit. The importance that e-businesses have gained in the last years and also its potential have convinced many companies to embrace this method when completing their activity totally or partially. Under the circumstances the auditors mission becomes more and more complex, being forced to handle equally both, the specific verifications for financial activity and the information systems used, involving in this case the use of various and complex technologies. This study presents the audits phases and their approach in e-businesses environment.

Keywords. E-business, Audit, Risk, Information Technology, Control, Auditor, Mission Planning

10.1 Introduction

The technological development led to major changes in business process at the end of XX century and the beginning of the XXI, pointed mainly in the use at a large scale of the virtual business environment. This is proven to be the right direction for moving forward, to evolve, allowing companies to reduce their expenses, to intensify the activity's rhythm, to increase their products and services' quality gaining new markets and though to become global. These changes have determined, necessarily, modifications at level

of verifying the operations' legality, accuracy and precision, respectively their audit. Considering the statement above, on one half the audit of information systems became needful, to verify the information flows, the systems and applications used during the execution, and, on the other half, but highly related, the audit of the correctness of information in documents, independently of their format: paper support or electronic. It is obvious that technological progress stimulated not only the software developers and the end-users, but also the auditors, which aren't able to fulfil their daily mission without understanding the way a company uses information technology to achieve its current activity. (Nearon 2001) The development of software applications like: Enterprise Resource Planning, Customer Resource Management, Supply Chain Management or the Internet, e-commerce and e-business determined a project for a new financial-accounting software system conceived and employed in a virtual environment, resumed by the concept of digital accounting. The mission of this new type of system is not only record and trace transaction information instantaneously but also cross-check internal and external documents automatically. (Rezaee and Reinstein 1998) But in the same time, the auditors had to manage an important volume of information to be checked, resulting the need of using specialized applications able to assist them during the audit verifications. All the changes mentioned above concurred to augment the specialists' interest to define new audit standards and to establish the specific activities in order to create the corresponding legislative environment (AICPA and CICA 1999 providing the required security for online transactions and their adequate check, including domain specific devices. (Yu et al. 2000; Helms 2002) We have to mention that there are many cases when the companies' managers do not understand completely all the aspects related to e-business and the risks involved. The situation is more critical for the auditors having less time to outline an opinion according to the reality regarding the company's activity, involving the analysis of a massy information volume delivered from various departments, developed in different environments and saved in different formats. One of the most debated solutions of the last period of time is the continuous audit – allowing a continuous evaluation and supervision of checking's, operations and risks. (Hunton et al. 2004; Alles et al. 2002) The over all impact of these complex and higher plane technologies is now visible on the modern day information system auditing processes and requires a relatively higher level of understanding by an auditor to succeed in the job. (Pathak 2003). It is necessary in these days for the auditor to have technical knowledge in order for him to accomplish his mission, confronted with new types of fraud, aspects related to the electronic signature, virtual transactions, with-

out paper support leading to changes in the audit specific activities like evaluating risks, modifications at the control level and specific tests.

10.2 Impacts of e-business on audit missions and auditors

As mentioned above, the development of e-business applies for a reconsideration of the way to accomplish the audit, respectively a combination between the problems specific to the software systems' audit and the ones related to financial audit. This requires from the auditors extensive knowledge regarding both the software systems involved and the information need to be mentioned in the financial-accounting documents, resumed in the financial statements at the end of the period. This study has the purpose to provide an answer for some elemental questions in this field like:

1. How is e-business affecting the financial audit of companies?
2. What aspects need to be followed by the auditors and what knowledge is requested in order to be solved?
3. Which are the audit risks in e-business environment?
4. What stages of an audit are experiencing changes when completing transactions in a virtual environment?

The answer to these questions will help the auditors to determine the problems that might appear during accomplishing audit missions and to improve the necessary skills suitable to the evolution of business environment in the XXI century.

10.2.1 Knowledge of the client's business and planning mission

The online environment presents a mix of technological, human and physical elements and threats to business can arise from any of these elements. (Deshmukh 2006) In Figure 10.1 it is presented the infrastructure for e-business and the connections between its components.

Knowledge the client business and planning the mission are the first stage in auditing. It involves an analysis of company's financial activities, also of information system's components participating to its achievement (hardware, software, human resources etc.) and of the company's business environment. „Knowledge of the Business” is referring to both – audited company's effective activity and the activity's environment. In the case of

using practices specific for e-business, the auditor must consider the changes generated by their employment and the related risks proportionally with their effects on company's financial situations. At this phase, the auditor must consider the following elements (IAPC 2002):

- The activity branch and field for the audited company.
- E-business strategy.
- The extensions of e-business activities.
- External service suppliers of the company.

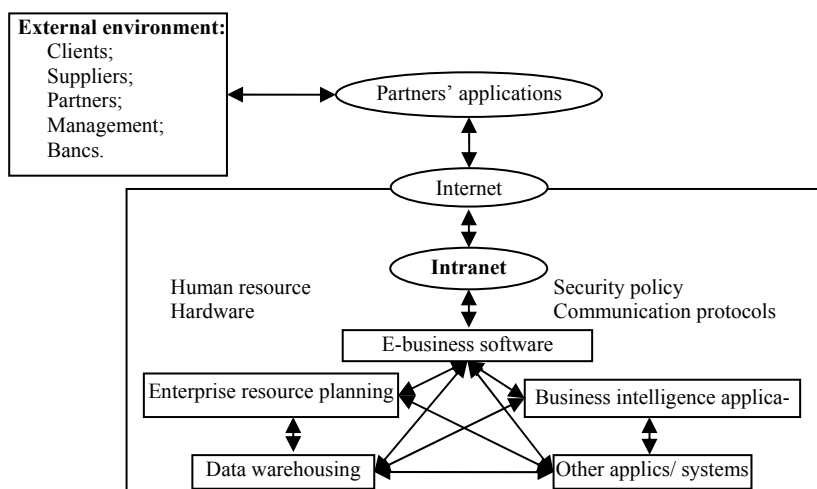


Fig. 10.1. E-business infrastructure

One of the essential aspects for online transactions having consequences for both –risks evaluation and verification procedures, is determined by immixture of other participants. Under the circumstances of using the Internet, the transactions are completed through a supplier of such services, or through a virtual, private network (Yahaya 2002) which might determine the auditors to request information from the specific supplier in order to establish exactly how is the information circulating in the system, the viability and also the rapport of respecting the legal provisions, the security policy established for each partner involved.

Considering the complexity and the plurality of used technologies, the different human resources employed in e-businesses, in planning the mission stage and knowledge the client business, the auditor must complete verifications at least for hardware equipment, the communication network components, both internal and services suppliers, supports and environ-

ments used to save data, software components, human resources employed.

To determine if a technology professional is needed, an audit planner is expected to consider these factors: complexity and usage of the firm's systems, significance of change to existing systems; extent of data sharing; extent of the entity's participation in e-business; audit evidence available in digital form. (Pathak and Lind 2003) There are cases when he may consider he is not having enough information technology competence to complete verifications and to provide an opinion according to the reality. In this case, the auditor may call for one or more experts in the information technology field.

The business environment of the company's activity has an important role in „knowledge the client business” stage because the capacity to continue a business depends not only on its members skills, but also on the market where the services and products are sold.

Specific to e-business, for the external environment evaluation and company's capacity to integrate, is estimating the audited company's resources to adjust to the technological development, from the retail products and services point of view, but also from the one of technologies used (hardware and software equipments) when contacting business partners, essential to perform business contacts in real-time on a global market, extremely dynamic, with no geographical limits, with less restrictions and where it is vital to trust the business partner.

10.2.2 Evaluating audit risks in e-business

Completing e-business audit also demands to reconsider the risks to be evaluated by the auditors in order to determine the work extension and to establish the size of the verified samples. The approach might be considered from two interdependent points of view: on one hand the risks related to the information system, and on the second one the risks related to the financial audit (Figure 10.3).

If the risks related to the information system are to be considered, the state is that for the e-business the emission speed is much higher confronted to the traditional business, due to inexistent borders in cyberspace. There are more classifications of e-business risks as internal and external, human or machine generated, intended or not. Some authors have identified three main categories of risks: technology related risks, relational risks and general risks. (Ratnasingam 2003)

Technology related risks. These are associated with the access to the e-commerce infrastructure. Their presence might cause deviations from the

performance targets expected in an electronic exchange „relationship”. For instance both the online seller and his partners may suffer hacker attacks. The security gap resulted from hacker attack may lead to losing income or „image” for the seller. On the other hand, the information circulating between partners has vulnerable points: internal applications of partner companies, application interface, network connections, mail boxes etc. If these vulnerable points are operated it might reach to presenting no-updated information or misinformation of clients and, even if the situation interferes for short period of time, they may sue the seller or ask for financial compensations.

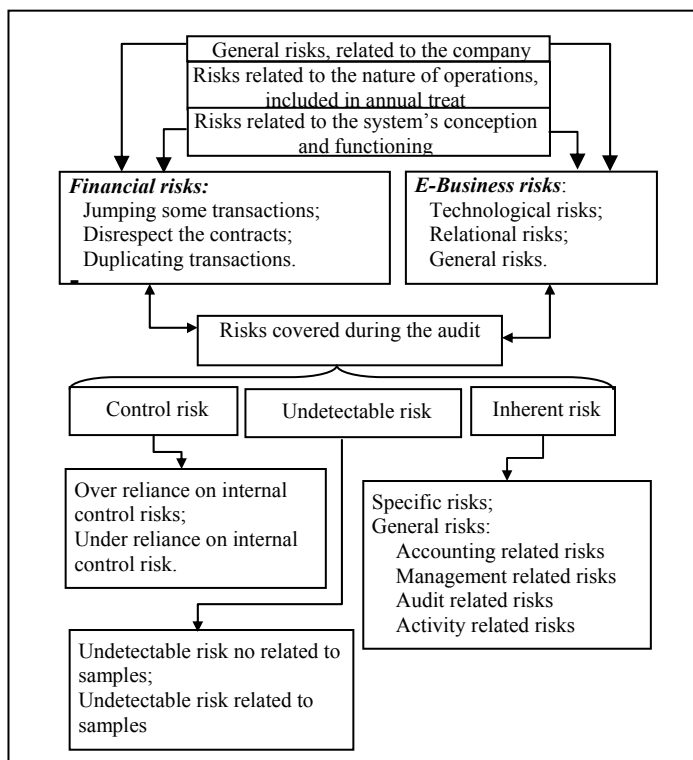


Fig. 10.2. Audit risks in e-business

Relational risks - are diverted from the lack of trust between business partners. Among their causes can be counted:

- The lack of experience and technical knowledge about e-business security.
- Opportunist behaviour.
- Resistance to changes.

- Ignorance concerning the method to audit the e-business activities.
- The uncertainty in the business environment.
- Ignorance concerning the methods to record (archive, memorize) the electronic transactions etc.

Examples of risks in this category: delay in production process, interrupted cash-flows, profit loss; all these might affect the predicted income and the business continuity.

General risks are referring to:

- Low quality business procedures.
- Environment risks.
- Lack of policy and standards.

Regarding the risks related to financial audit, they can be classified as following: general risks, company's specific risks, risks related to the nature of operations treated through annual accounts, risks related to the system's conception and function and risks undiscovered during the audit.

The general risks, specific for the company are affecting on the whole its operations. The auditor will analyze the company's activities and characteristics in order to get a better understanding of the potential risks able to take place and to identify the possible ones.

The risks related to the nature of operations treated through annual accounts are considering the manner of recording the three main data categories that the company is dealing: reiterative data resulting from daily activity, punctual data complementary to the reiterative ones, but distinguished at certain time periods, exceptional data, resulting from operations or decisions not specific to the daily activity: re-evaluations, mergers, reorganizations.

Risks related to the system's conception and functions are reported to conceiving the financial-accounting system so it allows detecting and correcting the errors, in serviceable time, especially through internal audits.

Risks undiscovered during the audit are affected by the auditor's choices regarding the procedures to use, the verifications' extension and dates.

There are risks mixing the two categories and crossed verifications are needed through the audited company-communication services suppliers-partners (customers, suppliers, banks, management etc). This range of risks determines the following measures to prevent/ control the consequences:

- Verifying the customers and suppliers' identity.
- Assure the transactions' integrity: validate the entries, prevent duplications or transactions' overlooks, respecting the contracts terms, prevent the transactions' partial processing, the correct information transfer be-

tween involved systems, assure the back-up copies and secure the system.

- Establish compelling transaction terms regarding the deliveries, payments, cash, products and services reception, giving and receiving credits etc.
- Establish security policies in order to assure the information confidentiality and protection, protocols between the involved parties.

10.2.3 Specific controls in auditing e-business

Executing verifications during an e-business auditing follows some classical attributes, regulated through audit standards. There can be identified from this point of view the following two verification types: general controls and application controls (Figure 10.3).

General controls regard the problems related to the fluent operating of the information system, including the systems' management, software programs' purchase and maintenance, physical and access security of equipments, programs and any saved data, planning the back-ups, conceiving and employing control devices integrated into the informatics equipments. (Arens et al. 2003) The general controls are pursuing two functions: on the one side the equipments and human resources management function, and on the other side the systems' proper functioning. In the e-business case, not only the internal function of information system has the important role, but also the external connection and this involves network audit – how the security standards are applied and the capacity to provide information from inside and outside the company.

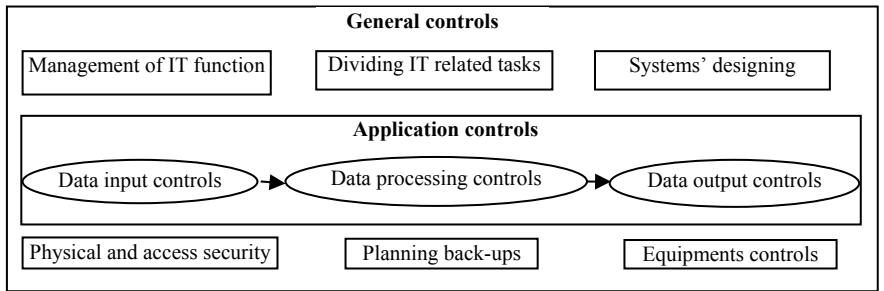


Fig. 10.3. Controls categories in e-business

Application controls track particular operations' verifications and must be executed for each account or operation category simulated by a software application. In the same category are also included the verifications

for each program or module used to realize online transactions: sales, purchase, payments, cash, human resources management etc.

In the virtual business environment it is significant for the transaction generated from the company's website to be correctly processed inside the company's internal system (Pathak 2005), for instance the accounting one, the customers and supplier's relations management, the state duties management. There are cases when this one is not integrated onto the internal system and must carefully follow the transactions' capture and transfer in the accounting system since it's affecting (IAPC 2002):

- The completeness and accuracy of transactions processing and information storage.
- The timing of the recognition of sales revenues, purchases and other transactions.
- Identification and recording of disputed transactions.

Auditing e-business assumes more compelling controls of the hardware and software components, of the communication realized inside the company and with external partners and also of the human resources employed to manage these components and activities.

Hardware equipments are solutions used by dedicated systems during executing online transactions and sometimes are requiring specific technologies, concurring to the operations' accomplishment successfully, their analyze must be done for each one separately, but also from the point of view of interactions between them.

Network components are the resistance piece for completing online transactions considering that they are based on network communication through Internet, Intranet and Extranet. At this level, the auditors must verify both the transactions accuracy and the execution speed, to identify possible intrusions, to verify the access rights and the degree of respecting them.

Saving supports where recorded clients, suppliers or other partner's data are, must guarantee to keep the information in proper conditions in order to be protected against natural dangers or human errors. With this purpose, companies must assure the execution and the storage in optimal conditions for original data and back-ups, especially when it comes to important information.

Software components are referring to both operation systems and specific applications (Enterprise Resource Planning, databases, e-business applications etc.). The audit must aim in this case each application, independently, but also the interaction between modules, the accuracy and the exhaustive of the data transfers.

Human resource is an essential component for any company, for both: the one charged with information technologies' development and maintenance and the other involved in financial or commercial transactions. The auditor must identify the participants to online transactions, to verify their capacity to complete the tasks, to recommend, eventually, training sessions.

10.3 Solutions for an efficient financial auditing in e-business

The e-business audit is a critical activity due to its complexity and to the diversity of information technologies and also of online transactions. In many cases the participants are coming from various environments, with different laws and standards. In order to increase the audit's efficiency in this field the following measures' categories are recommended:

- Unifying the legal settlements in the financial-accounting domain and the information technologies.
- Continuous trainings about information technologies for the auditors. In this case, a solution might be to ask for different auditors, information technologies specialists, but it might also increase the time needed to complete an audit mission and the risks related to the skill to define an objective opinion according to the reality.
- Standardization of equipments and applications used to manage the company's activity and to establish contacts and contracts with the external environment, hard to accomplish due to the existent technologies diversity and fast development.

Building applications able to continuously supervise the activity, which need to be analyzed with attention by the external auditors.

In the present days it is common either to ask for auditors which are specialized in information technologies, either to specialized trainings for the existent auditors. Another solutions, often applied, is to separate the information system auditing from the financial audit, which might lead to spreading the errors or to conflicts.

10.4 Conclusion

This study wanted to identify the main changes taking place during a financial auditing of e-business. Here are presented some conclusions came off from above:

- The extension of e-business to a global scale induces changes in all financial audits' phases.
- In order to reply to the market requests, the auditors' knowledge in the financial accounting domain has become insufficient, being necessary to add information technologies' capabilities related to e-business operations.
- The information technologies' function in the business process is essential and in the same time is conditioning the business' success in the virtual environment.
- The failure or the success of an audit mission is depending, in all the situations, on the commitment of the company's management and employers for its accomplishment and also on the auditors' skills and knowledge.

There is need to be mentioned that the present study shows only a few of the changes appearing, during the audit missions, when passing from the classical approach for doing business to the virtual environment.

References

1. AICPA, CICA (1999) Electronic Commerce Assurance Services Task Force. WebTrust Principles and Criteria for Business-to-Consumer Electronic Commerce, Version 1.1
2. AICPA, CICA (1999), Electronic Commerce Assurance Services Task Force. WebTrust-ISP Principles and Criteria for Internet Services Providers in Electronic Commerce, Version 1.0
3. Alles M, Kogan A, Vasarhelyi M (2002) Feasibility and economics of continuous assurance. *Auditing: A Journal of Practice & Theory* 21:125-138
4. Arens A, Loebbecke J, Elder J, Beasley S (2003) *Auditing: An integrated approach*. Arc Publishing House, Chisinau
5. Deshmukh A (2006) *Digital accounting. The Effects of the Internet and ERP on Accounting*, IRM Press, London
6. Helms G (2002) Traditional and emerging methods of the electronic assurance. *CPA Journal*, pp 26-31
7. Hunton EJ, Wright MA, Wright S (2004) Continuous reporting and continuous assurance: Opportunities for behavioral accounting research. *Journal of Emerging Technologies in Accounting* 1:91-102

8. IAPC (2002) IAPS 1013 - Electronic commerce: Effect on the audit of financial statements. http://www.paab.co.za/documents/doc_00301.pdf
9. Nearon HB (2001) Auditing e-business. *The CPA Journal*
10. Pathak J (2003) Model for audit engagement planning of e-commerce. *International Journal of Auditing* 2:121-142.
11. Pathak J (2005) *Information technology auditing. An evolving agenda.* Springer-Verlag, Berlin
12. Pathak J, Lind R (2003) Audit risk, complex technology, & auditing processes. *EDPACS* 5: 1-9
13. Ratnasingam P (2003) *Inter-organizational trust for business-to-business e-commerce.* IRM Press, London
14. Rezaee Z, Reinstein A (1998) The impact of emerging information technology on auditing. *Managerial Accounting Journal* 13:465-471
15. Yahaya Y, Little D, Onuh S (2002) Multiple case studies of total enterprise integration programs: lessons and benefits. *International Journal of Manufacturing Technology and Management* 3/4:283-302
16. Yu C, Yu H, Chou C (2000) The impacts of electronic commerce on auditing practices: An auditing process model for evidence collection and validation. *International Journal of Intelligent Systems in Accounting, Finance & Management* 3:195-216

Chapter 11

Reputed authenticated routing ad-hoc networks protocol

Ahmed Sameh, Abdalla Mahmoud, Sherif El-Kassas

Department of Computer Science, The American University in Cairo, P.O.Box 2511, Cairo, EGYPT, Email: sameh@aucegypt.edu

Abstract. In this paper, we analyze one of the secure mobile ad hoc networks protocols, which is Authenticated routing for ad hoc networks (ARAN). Such protocol is classified as a secure reactive routing protocol, which is based on some type of query-reply dialog. That means ARAN does not attempt to continuously maintain the up-to-date topology of the network, but rather when there is a need, it invokes a function to find a route to the destination. Here, we detail the security attacks that the ARAN protocol defends against, criticize how an authenticated selfish node can disturb the network by dropping packets or by not cooperating in the routing functionality and propose a reputation-based scheme called Reputed-ARAN to detect and defend against selfish nodes.

Keywords. Mobile Ad hoc Networks, Routing Security, ARAN, Malicious nodes, Selfish nodes, Reputation-based Schemes

11.1 Introduction

The Authenticated Routing for Ad Hoc Networks (ARAN) secure routing protocol proposed by Sanzgiri, Laflamme, Dahill, Levine, Shields and Belding-Royer [1] is seen as an on-demand routing protocol that detects and protects against malicious activities caused by other nodes and peers in the ad hoc environment. This protocol introduces authentication, message

integrity and non-repudiation as part of a minimal security policy for the ad hoc environment. Our evaluation of ARAN shows that it has minimal performance costs for the increased security in terms of processing and networking overhead. However, by analyzing the protocol thoroughly, we figured out that an authenticated selfish node can disrupt the network by promising to cooperate in the routing functionality and then does not participate at all. This results in wasting the mobile ad hoc network bandwidth.

11.2 Security analysis of ARAN

In this section, we will delve into two types of mobile ad hoc networks (MANETs) nodes: Malicious and selfish nodes. Then, we present a security analysis of ARAN protocol describing how it faces most of the malicious nodes attacks and fails to face selfish nodes.

Malicious nodes can disrupt the correct functioning of a routing protocol by modifying routing information, by fabricating false routing information and by impersonating other nodes. On the other side, selfish nodes can severely degrade network performances and eventually partition the network by simply not participating in the network operation [2]. Current Ad Hoc routing protocols do not address the selfishness problem and assumes that all nodes in the MANET will cooperate to provide the required network functionalities. In this section, we analyze the Authenticated Routing for Ad Hoc Networks by evaluating its robustness in the presence of the different attacks introduced in earlier subsection [1].

- Unauthorized participation: Since all ARAN packets must be signed, a node can not participate in routing without authorization from the trusted certificate server. This access control therefore rests in the security of the trusted authority, the authorization mechanisms employed by the trusted authority, the strength of the issued certificates, and the revocation mechanism.
- Spoofed Route Signaling: Route discovery packets contain the certificate of the source node and are signed with the source's private key. Similarly, reply packets include the destination node's certificate and signature, ensuring that only the destination can respond to route discovery. This prevents impersonation attacks where either the source or destination node is spoofed.
- Fabricated Routing Messages: Since all routing messages must include the sending node's certificate and signature, ARAN ensures non-

repudiation and prevents spoofing and unauthorized participation in routing.

- **Alteration of Routing Messages:** ARAN specifies that all fields of RDP and RREP packets remain unchanged between source and destination. Since both packet types are signed by the initiating node, any alterations in transit would be detected, and the altered packet would be subsequently discarded. Thus, modification attacks are prevented in ARAN.
- **Denial-of-Service Attacks:** Denial-of-service (DoS) attacks can be conducted by nodes with or without valid ARAN certificates. In the certificate-less case, all possible attacks are limited to the attacker's immediate neighbors because unsigned route requests are dropped. However, nodes with valid certificates can conduct effective DoS attacks by sending many unnecessary route requests and they will go undetected as the current existing ARAN protocol can not differentiate between legitimate and malicious RREQs coming from authenticated nodes.

It is clear from the above mentioned security analysis of the ARAN protocol that ARAN is a secure MANET routing protocol providing authentication, message integrity, confidentiality and non-repudiation by using certificates infrastructure. As a consequence, ARAN is capable of defending itself against spoofing, fabrication, modification, DoS and disclosure attacks. However, erratic behavior can come from a malicious node, which will be defended against successfully by existing ARAN protocol, and can also come from an authenticated node. The currently existing ARAN secure routing protocol does not account for attacks that are conducted by authenticated selfish nodes as these nodes trust each other to cooperate in providing network functionalities. This results in that ARAN fails to detect and defend against an authenticated selfish node participating in the mobile ad hoc network. Thus, if an authenticated selfish node does not forward or intentionally drop control or data packets, the current specification of ARAN routing protocol can not detect or defend against such authenticated selfish nodes. This weakness in ARAN specification will result in the disturbance of the ad hoc network and the waste of the network bandwidth. In the following section, we are proposing a solution to account for this type of attack.

11.3 The proposed reputation system scheme: Reputed-ARAN

Performance of Mobile Ad Hoc Networks is well known to suffer from free-riding as there is a natural incentive for nodes to only consume, but

not contribute to the services of the system [3]. So in the following subsections, we will introduce our newly designed reputation system scheme that we plan to integrate with the ARAN routing protocol, ending up having Reputed-ARAN.

Whereas most of the attacks performed by malicious nodes can be detected and defended against by the use of the secure routing ARAN protocol, as was explained earlier, there remain the attacks that an authenticated selfish node can perform.

There are two attacks that an authenticated selfish node can perform that the current ARAN protocol can not defend against. To illustrate these two possible attacks that a selfish node can use to save its resources in a MANET communication, the attack-tree notation proposed by Bruce Schneier [4] that allows the categorization of attacks that lead an attacker to reach a specific goal is used. In the below table, the attack tree that can not be detected by current ARAN protocol is shown:

Table 11.1. Attack Tree: Save own resources

Attack tree: Save own resources	
OR	1. Do not participate in routing
	1. Do not relay routing data
	OR
	1. Do not relay route requests
	2. Do not relay route replies
	2. Do not relay data packets
	1. Drop data packets

As shown in the above table, when nodes simply drop packets (case 1.1 and 2.1 in the attack tree), all the security features of ARAN fail to detect or defend against these attacks, as they focus only on the detection of malicious nodes' attacks and not the authenticated selfish nodes' attacks. ARAN protocol assumes that authenticated nodes are to cooperate and work together to provide the routing functionalities.

In this section, we are proposing a simple reputation based scheme to mitigate misbehavior and enforce cooperation in MANETs using ARAN protocol. We call this newly designed system: Reputed-ARAN. Different from global reputation based schemes like Confidant and Core, discussed in [5] and [6], our solution uses local reputation only. Each node keeps only the reputation values of all its one-hop neighbors. Our work is partially following the same methodology that Prashant, Partha and Amiya used in their paper about reputation systems for AODV [7].

In the proposed reputation scheme, all the nodes in the mobile ad hoc network will be assigned an initial value of null (0) as in the Ocean reputation-based scheme [8]. Also, the functionality of the normal ARAN

routing protocol in the authenticated route setup phase will be modified so that instead of the destination unicasts a RREP to the first received RDP packet of a specific sender only, the destination will unicast a RREP for each RDP packet it receives and forward this RREP on the reverse-path. The next-hop node will relay this RREP. This process continues until the RREP reaches the sender. After that, the source node sends the data packet to the node with the highest reputation. Then the intermediate node forwards the data packet to the next hop with the highest reputation and the process is repeated till the packet reaches its destination. The destination acknowledges the data packet (DACK) to the source that updates its reputation table by giving a recommendation of (+1) to the first hop of the reverse path. All the intermediate nodes in the route give a recommendation of (+1) to their respective next hop in the route and update their local reputation tables. If there is a selfish node in the route, the data packet does not reach its destination. As a result, the source does not receive any DACK for the data packet in appropriate time. So, the source gives a recommendation of (-2) to the first hop on the route. The intermediate nodes also give a recommendation (-2) to their next hop in the route up to the node that dropped the packet. As a consequence, all the nodes between the selfish node and the sender, including the selfish node, get a recommendation of (-2). The idea of giving (-2) to selfish nodes per each data packet dropping is due to the fact that negative behavior should be given greater weight than positive behavior. In addition, this way prevents a selfish node from dropping alternate packets in order to keep its reputation constant. This makes it more difficult for a selfish node to build up a good reputation to attack for a sustained period of time [9]. Moreover, the selfish node will be isolated if its reputation reached a threshold of (-40) as in the Ocean reputation-based scheme [8]. In the following table, the default Reputed-ARAN parameters are listed:

Table 11.2. Reputed-ARAN Default parameters

Initial Reputation	0
Positive Recommendation	+1
Negative Recommendation	-2
Selfish drop Threshold	-40
Re-induction timeout	5 minutes

The proposed protocol will be structured into the following four main phases, which will be explained in the subsequent subsections: Route Lookup Phase, Data Transfer Phase, Reputation Phase, and Timeout Phase. This phase mainly incorporates the authenticated route discovery and route setup phases of the normal ARAN secure routing protocol. In this phase, if

a source node S has packets for the destination node D, the source node broadcasts a route discovery packet (RDP) for a route from node S to node D. Each intermediate node interested in cooperating to route this control packet broadcasts it throughout the mobile ad hoc network; in addition, each intermediate node inserts a record of the source, nonce, destination and previous-hop of this packet in its routing records. This process continues until this RDP packet reaches the destination. Then the destination unicasts a route reply packet (RREP) for each RDP packet it receives back using the reverse-path. Each intermediate node receiving this RREP updates its routing table for the next-hop of the route reply packet and then unicasts this RREP in the reverse-path using the earlier-stored previous-hop node information. This process repeats until the RREP packet reaches the source node S. Finally, the source node S inserts a record for the destination node D in its routing table for each received RREP.

In the below figures, the route lookup phase is presented in details, illustrating the two phases of it, the authenticated route discovery phase and the authenticated route setup phase.

Next Hop	Reputation	Next Hop	Reputation
A	10	C	20
B	5	E	-5

Fig. 11.1. A MANET Environment

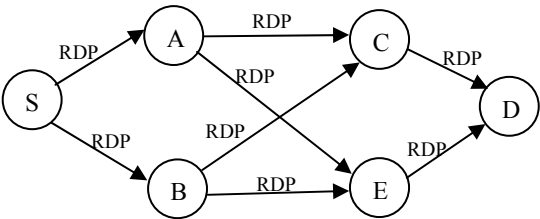


Fig. 11.2. Broadcasting RDP

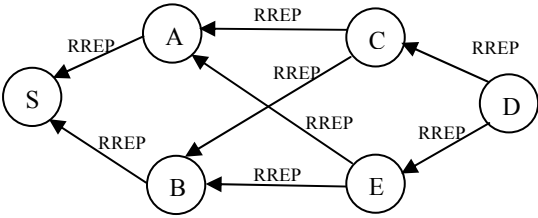


Fig. 11.3. Replying to each RDP

At this time, the source node S and the other intermediate nodes have many RREPs for the same RDP packet sent earlier. So, the source node S chooses the highly-reputed next-hop node for its data transfer. If two next-hop nodes have the same reputation, S will choose one of them randomly, stores its information in the sent-table as the path for its data transfer. Also, the source node will start a timer before it should receive a data acknowledgement (DACK) from the destination for this data packet. Afterwards, the chosen next-hop node will again choose the highly-reputed next-hop node from its routing table and will store its information in its sent-table as the path of this data transfer. Also, this chosen node will start a timer, before which it should receive the DACK from the destination for this data packet. This process continues till the data packet reaches the destination node D. And of course in this phase, if the data packet has originated from a low-reputed node, the packet is put back at the end of the queue of the current node. If the packet has originated from a high-reputed node, the current node sends the data packet to the next highly-reputed hop in the route discovered in the previous phase as soon as possible. Once the packet reaches its destination, the destination node D sends a signed data acknowledgement packet to the source S. The DACK traverses the same route as the data packet, but in the reverse direction.

In the following figure, the data transfer phase is illustrated:

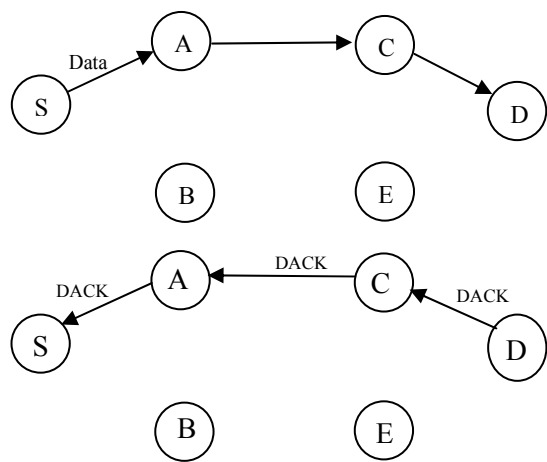


Fig. 11.4. Sending Data Acknowledgement for each received data packet

In this phase, when an intermediate node receives a data acknowledgement packet (DACK), it retrieves the record, inserted in the data transfer phase, corresponding to this data packet then it increments the reputation of the next hop node. In addition, it deletes this data packet entry from its

sent-table. Once the DACK packet reaches node S, it deletes this entry from its sent-table and gives a recommendation of (+1) to the node that delivered the acknowledgement.

In this phase, once the timer for a given data packet expires at a node, the node retrieves the entry corresponding to this data transfer operation returned by the timer from its sent-table. Then, the node gives a negative recommendation (-2) to the next-hop node and deletes the entry from the sent-table. Later on, when the intermediate nodes' timers up to the node that dropped the packet expire, they give a negative recommendation to their next hop node and delete the entry from their sent-table. As a consequence, all the nodes between the selfish node and the sender, including the selfish node, get a recommendation of (-2). Now, if the reputation of the next-hop node goes below the threshold (-40), the current node deactivates this node in its routing table and sends an error message RERR to the upstream nodes in the route. Then the original ARAN protocol handles it. Now, it is the responsibility of the sender to reinitiate the route discovery again. In addition, the node whose reputation value reached (-40) is now temporally weeded out of the MANET for five minutes and it later joins the network with a value of (0) so that to treat it as a newly joined node in the network.

11.4 Simulation

This section will be presenting the simulation environment, metrics and the results of the comparison between the proposed Reputed-ARAN and normal ARAN protocol.

Table 11.3. Configuration values that are used in all the performed simulations

Parameter	Value
MANET Area	670*670 m ²
Total number of nodes	20
% of Selfish Nodes Simulated	0% up to 30%
Movement Pattern	Random Waypoint
Node Speed	0 up to 10 m/s
Application	Five Constant Bit Rate (CBR)
Number of generated Packets	1000 packets per CBR
Size of Packet	512 bytes
Simulation Time	15 minutes

All simulation experiments are developed and simulated on an Intel 1.4 GHz machine using Linux Red Hat 9.0 with 256 MB RAM and the net-

work simulator Glomosim [10], a library-based sequential and parallel simulator for wireless mobile ad-hoc networks. The Table 11.3 summarizes the different configuration values that are used in all the performed simulations.

In order to compare the performance of Reputed-ARAN and normal ARAN, both protocols are run under identical mobility and traffic scenarios. First, an analysis of normal well-behaved ARAN network is done. Then, some uncooperative nodes are introduced to the normal ARAN network and analysis of the performance is done. Following that, the newly designed reputation-based scheme is added to normal ARAN, to become Reputed-ARAN, and performance is compared to normal ARAN. A comparison between both of these protocols using the metrics explained is presented in the following subsections.

This value represents the ratio of the total number of packets that reach their destination, to the total number of packets sent by the source. It is calculated according to this formula: $\text{Throughput} = \text{Packets Received} / \text{Packets Sent}$. The network throughput is directly influenced by packet loss, which may be caused by general network faults or uncooperative behavior.

This is the average delay between the sending of the data packet by the constant bit rate source and its receipt at the corresponding constant bit rate receiver.

In this experiment, the network throughput is being measured for the normal ARAN secure routing protocol and the Reputed-ARAN. The node speed and the percentage of selfish nodes participating in the mobile ad hoc network are varied to compare the results.

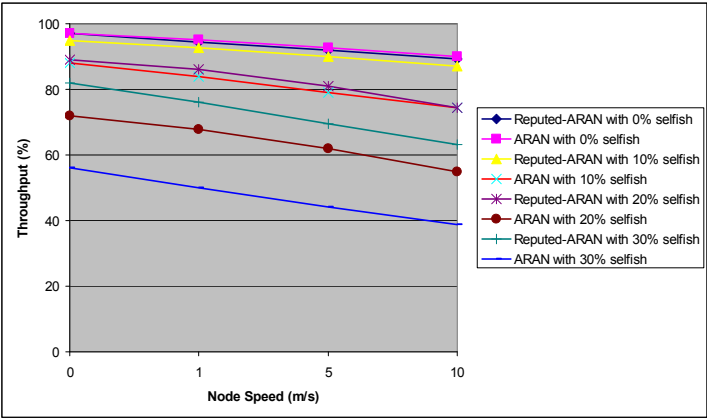


Fig. 11.6. Effects of Selfish nodes on Network Throughput

From the above graph and its corresponding table, it is clear that the lack of cooperation has fatal effect on the efficient working of the MANET. This graph shows the dramatic fall in normal ARAN's network throughput with increasing percentage of selfish nodes. The different curves show a network of 20 nodes with different percentages of selfish nodes, from 0% up to 30%, and moving at different speeds.

The below figure shows the results of the overhead metric of both protocols: normal ARAN and Reputed-ARAN with different node speed. From the above graph and its corresponding table, it is clear that the newly proposed Reputed-ARAN protocol has a higher overhead than the normal ARAN secure routing protocol.

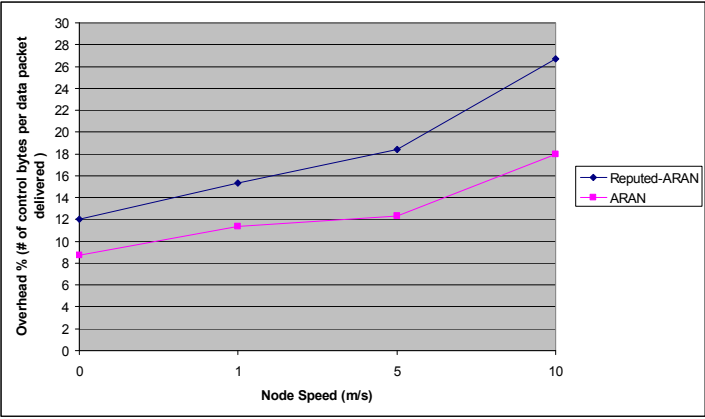


Fig. 11.7. Overhead Percentage

These extra RREPs are used later for the choice of the highly-reputed next-hop node in the data transfer phase. Thus, when nodes are moving at speed of 10 m/s, the overhead percentage rises from 18%, in case of normal ARAN, to 26.7%, in case of Reputed-ARAN. Though the overhead percentage added by the Reputed-ARAN is significant, this reputation-based scheme still improves considerably the network throughput, as shown in the previous section. In this experiment, the average end-to-end delay of data packets for the normal ARAN secure routing protocol and the Reputed-ARAN is measured. The percentage of selfish nodes participating in the mobile ad hoc network is varied to compare the results. The below figure shows the results of the average end-to-end delay of data packets metric of both protocols: normal ARAN and Reputed-ARAN with different percentage of selfish.

11.5 Conclusion and future directions

According to the many simulations that were performed, the newly proposed reputation-based scheme, built on top of normal ARAN secure routing protocol, achieves a higher throughput than the normal ARAN in the presence of selfish nodes. Thus, the proposed design, Reputed-ARAN, proves to be more efficient and more secure than normal ARAN secure routing protocol in defending against both malicious and authenticated selfish nodes. As for future work, there are many open-problems in the ad hoc networks field such as Quality of Service and power-aware routing protocols that need to be studied and researched in depth.

References

1. Sanzgiri Kimaya, Laflamme Daniel, Dahill Bridget, Neil Levine Brian, Shields Clay, Belding-Royer Elizabeth M (2005) Authenticated routing for ad hoc networks. MobiCom.
2. Michiardi P, Molva R (2002) Simulation-based analysis of security exposures in mobile ad hoc networks. In: Proceedings of European Wireless Conference
3. Dewan P, Dasgupta P (2003) Trusting routers and relays in ad hoc networks. In: First International Workshop on Wireless Security and Privacy in conjunction with IEEE International Conference on Parallel Processing Workshops (ICPP)
4. Schneier Bruce (1999) Modeling security threats. Dr Dobb's Journal
5. Buchegger Sonja, Le Boudec Jean-Yves (2002) Performance analysis of the CONFIDANT protocol: Cooperation of Nodes, Fairness In Dynamic Ad-hoc NeTworks. In: Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing, MobiHOC, Lausanne, CH

Chapter 12

Quantum mechanical analysis of nanowire FETs

Alireza Kargar, Rahim Ghayour

Dept. of Elect. Eng., School of Eng., Shiraz University, Shiraz, Iran
E-mail: rghayour@shirazu.ac.ir

Abstract. In this paper, we present analysis of a coaxial Nanowire FET by combining the analytical and numerical approaches. We achieve electrical characteristics of cylindrical Nanowire FET and investigate the effects of changing metal/semiconductor Schottky barrier height and gate oxide thickness on electrical characteristics. First, we solve 1-D Laplace equation analytically to achieve the potential distribution within the wire. Then, we show the effect of changing the gate oxide thickness on the potential distribution. Second, by applying WKB approximation to tunneling problem and assuming ballistic current transport, we numerically achieve transfer characteristics of FET. Finally, we investigate the effect of varying Schottky barrier height (SB) on electrical characteristics of Nanowire FET.

Keywords. Coaxial NanowireFET, Schottky Barrier, Ballistic Transport, Potential Distribution, Gate oxide Thickness

12.1 Introduction

As the channel length of conventional planar metal oxide semiconductor field effect transistor shrinks into the nanometer scale, behavior of the device degrades mainly because of short channel effects due to weakened gate control. In the Nanowire field effect transistor (NWFET), quantum effects such as the tunneling and confinement effects should be considered

carefully to assess the device performance accurately. Also the ballistic transport can be regarded as making a dominant contribution to the total current. In particular, as the channel length of the NWFET shrinks, the source-to-drain direct tunneling current is expected to be increased significantly until it becomes the main source for the short channel effects in the NWFETs. On the other hand, semiconducting nanowire transistors are attracting attention due to their potential applications such as electronics [1] and bimolecular detection [2]. Also large ON-currents, large I_{ON}/I_{OFF} ratios have been reported [1]. Such devices are typically fabricated in the plane of a substrate with either a top or bottom gate. It is well known that a fully surround gate structure, is optimal for electrostatic control over charge carriers in the channel [3]. Further, most attempts on nanowire modeling assumed ballistic transport [4, 5].

In section 12.2, first, by using an analytical approach, we present the electrostatics potential distribution of coaxial nanowire FET. Then in ballistic transport regime and application of WKB approximation, we numerically offer transfer characteristics. In section 12.3, by looking at previous studies, we investigate the dependence of the carrier transport on the metal/semiconductor Schottky barrier height and gate oxide thickness.

12.2 Device structure and approach

Let us consider a coaxial gate NWFET as shown in Figure 12.1. The channel between source and drain is considered a one-dimensional wire, located along the axis of the structure. The bias voltage of the gate is applied to the outer shell of the structure. The space between wire and outer shell is filled with insulator that is known as the gate oxide geometry.

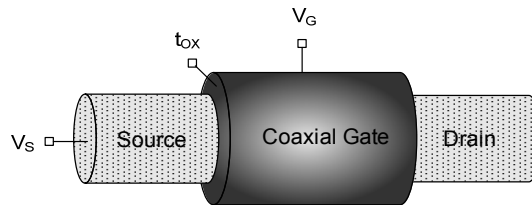


Fig. 12.1. Schematic view of the CNWFET device structure. The gate forms the curved surface of the outer cylinder.

The potential distribution of the structure is achieved by solving the Laplace's equation in cylindrical coordinates as:

$$\frac{1}{\rho} \frac{\partial}{\partial \rho} \left(\rho \frac{\partial U(\rho, z)}{\partial \rho} \right) + \frac{\partial^2 U(\rho, z)}{\partial^2 z} = 0 \quad (12.1)$$

where $U(\rho, z)$ is the electrostatic potential distribution along the nanowire. Charges in the nanowire are neglected in the simulation. Thus, solving this equation analytically we achieve the potential distribution which is in exponential form.

Figure 12.2 demonstrates the potential distribution. The end contacts in Nanowire FET are considered the reservoirs of charge, from which the carriers are injected into the nanowire. The amount of injection depends on the transmission coefficient and the drain-source voltage of a Nanowire FET. The electrons which are coming from the source reservoir into the nanowire experience an exponential shape barrier which is the Schottky barrier between the wire and the source metal end, where it is demonstrated by Figure 12.2. For a Schottky barrier, the Schrödinger equation can not be precisely solved, so the Wentzel-Kramers-Brillouin (WKB) approximation is often the most widely used approach. Using this approach the transmission probability can be derived as [6]:

$$T(E) = \exp \left(-2 \int_{z_i}^{z_f} \sqrt{\frac{2m_{eff}(U(z) - E)}{\hbar^2}} dz \right) \quad (12.2)$$

where $U(z)$ is the potential distribution, E is its total energy, $\hbar$ is reduced Planck's constant, m_{eff} is the carrier effective mass, and z_i and z_f are the classical turning points. So, we analytically calculate transmission probability from equation (12.2).

Assuming the ballistic transport, the channel current can be expressed using the Landauer-Büttiker formula as follows [7]:

$$I_{DS} = \frac{2q}{\pi \hbar} \int T(E)(f_S(E) - f_D(E)) dE \quad (12.3)$$

where f_S and f_D are the Fermi-Dirac functions with Fermi level of the source and the drain, respectively. From a numerical point of view, we calculate the electrical characteristics by means of a simple numerical method from equation (12.3).

12.3 Results

The gate oxide thickness (t_{OX}) in nanowire transistor is an important parameter that affects the performance of transistor and can affect the current of the transistor seriously. In fact, the electrostatic potential of the coaxial Nanowire FET (CNWFET) changes with different gate oxide thickness. Figure 12.2 shows the electrostatic potential distribution at the gate-to-source voltage $V_{GS}=0.2$ V and gate oxide thickness of $t_{OX}=2, 8, 20$ nm. It demonstrates that the electrostatic potential in ballistic transport of nanowire FET intensively vary with t_{OX} . When the gate oxide thickness is the same as the channel length in ballistic transport, the source/drain field penetrates into the channel and the transistor cannot be turned off. Also, when the gate oxide is thin, however, the gate still has very good control over the channel and the transistor is well turned off [8].

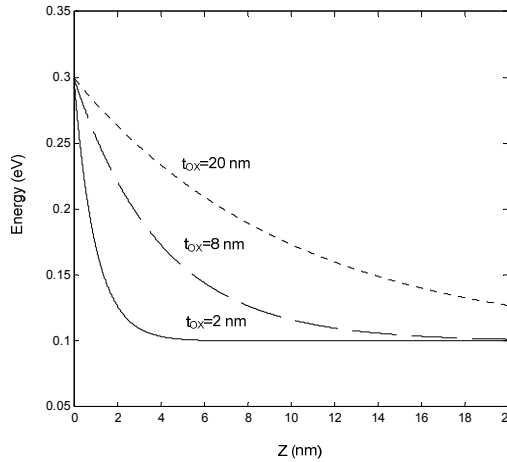


Fig. 12.2. The potential distribution in the z direction for different gate oxide thickness and gate-source voltage is equal to 0.2 eV.

In addition to gate oxide thickness, the metal/semiconductor Schottky barrier height (SB height) has a vital role in correct performance of nanowiretransistor so that a great increase/decrease of SB, caused the transistor can not be well turned on/turned off. In Figure 12.3 the transfer characteristics (I_{DS} - V_{GS} plot) are plotted for a drain-to-source voltage of $V_{DS}=0.3$ volt and different Schottky barrier heights (SB height).

Figure 12.4 shows the transfer characteristic, I_{DS} versus V_{DS} , with different V_{GS} , SB=0.3 eV and $t_{OX}=2$ nm so that it confirms, the fact which drain-source current increases with the larger amounts of gate-source voltage.

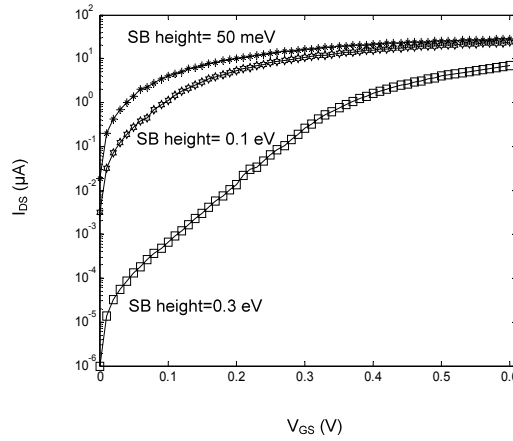


Fig. 12.3. The transferred curves I_{DS} - V_{GS} of the CNWFET with $V_{DS}=0.3$ V, SB height of 50 meV, 0.1 eV, 0.3 eV and $t_{OX}=2$ nm.

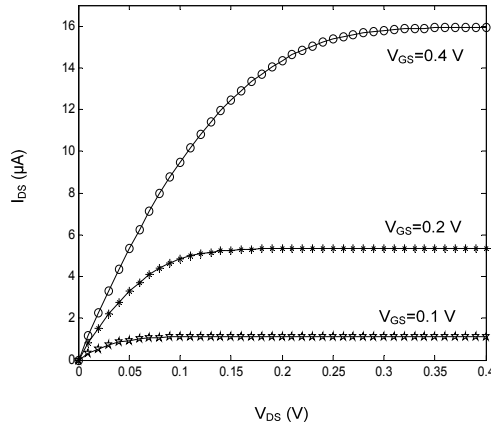


Fig. 12.4. The computed I_{DS} - V_{DS} curves of the CNWFET with $V_{GS}=0.1, 0.2, 0.4$ V and $t_{OX}=2$ nm.

12.4 Conclusion

The coaxial nanowire FET (CNWFET) is presented and by applying an analytical-numerical approach the electrical characteristics are achieved. The ballistic transport is assumed for carrier transport such as [5] and for handling the tunneling problem the WKB approximation is suited. So, we demonstrate that in ballistic regime for CNWFET, two essential factors are

the insulator thickness and Schottky barrier height so that, with optimization of these factors, the on-off current ratio and generally the electrical performance of this structure can be improved.

References

1. Xiang J, Lu W, Hu Y, Wu Y, Yan H, Lieber CM (2006) Ge/Si nanowire heterostructure as high-performance field-effect transistors. *Nature* 441:489-493
2. Hahn J, Lieber CM (2004) Label-free detection of DNA hybridization using carbon nanotube network field-effect transistors. *Nano Lett.* 103:921-926
3. Wang J, Polizzi E, Lundstrom M (2004) A three-dimensional quantum simulation of silicon nanowire transistors with the effective-mass approximation. *J. Appl.* 96:2192-2203
4. Chen J (2006) A circuit-compatible model for ballistic nanowire transistors. In: *Proceeding Nanotechnology Conference*, pp 850-853
5. Jimenes D, Saenz JJ, Iniguez B, Sune J, Marsal LF, Pallares J (2003) Unified compact model for the ballistic quantum wire and quantum well metal-oxide-semiconductor field-effect-transistor. *J. Appl. Phys.* 94:1061-1068
6. Powell and Crasman, *Quantum Mechanics*
7. Datta S (1998) *Electronic transport in mesoscopic systems*. Cambridge Univ. Press, Cambridge, U.K.
8. Guo J, Wang J, Pollizi E, Datta S, Lundstrom M (2003) Electrostatics of nanowire transistors. *IEEE Transaction on Nanotechnology* 2:329-334

Part 14

Software tools and programming languages

The purpose of this part is to provide a forum for software scientists to present and discuss recent researches and developments in the programming languages and software. The scope of the part is not restricted, it covers ongoing research in different countries related to software technology with emphasize in programming languages and software tools e.g.: algorithms and data structures, programming languages and paradigms, software engineering, information systems, logic and declarative programming, parallel and distributed computing, embedded computing, etc. The papers include the impact of quality factors on the success of software development, language structure decompositions, embedded applications for driverless metro train, a post-mortem JavaSpaces debugger, stochastic numerical simulations using C#, visualization and analyses of multi-dimensional data sets using OLAP, design of a computer-based triangle chess machine, a taxonomy model for analyzing project-based learning using analytical hierarchy process , a framework of mixed model of a neural network and boolean logic for financial crisis prediction, chinese document clustering using self-organizing map-based on botanical document warehouse, linear discriminant analysis in Ottoman alphabet character recognition, user requirements engineering and management in software development and tools for active filter synthesis.

Chapter 1

Impact of quality factors on the success of software development

M. Sepehri,¹ A. Abdollahzadeh,² M. Sepehri, M. Goodarzi

¹Department of Computer Engineering & IT, Islamic Azad University
Fars Sciences and Research Branch, Fars, IRAN

²Department of Computer Engineering, Amir Kabir University
Tehran, IRAN

Abstract. During the design of a system, software engineering is faced with many choices. Most of these choices involve the identification of a solution for a problem from a given set of alternatives. The alternative solutions generally only differ with respect to quality attributes. Quality factors have very strong impact on the success of software development methodologies, therefore the alternative should be selected which satisfies the quality requirements best. This paper was undertaken to gathering popular methodologies such as SSADM (Structured System Analysis and Design) and Object-Oriented towards the impact of quality factors on methodology choice. In this paper, an empirical comparison was made between two mentioned methodologies used with a software project. The mentioned methodologies for project domain were compared using some quality factors such as maintainability, flexibility, reliability and usability. The obtained results indicated that investigation of quality factors such as maintainability, flexibility, reliability in Object-Oriented methodology was achieved in less time than SSADM, whereas on usability parameter users were clearly more satisfied with their experience with the SSADM.

Keywords. SSADM Methodology, Object-Oriented Methodology, Maintainability, Reliability, Flexibility, Usability.

1.1 Introduction

Engineering approach for constructing systems needs methodology and suitable tools to produce software. There is no general software development methodology powerful enough to handle all the variations of today systems [11]. Methodology could be revised, improved and extended [6].

The main objective of this paper is to compare and evaluate two evaluation methodologies such as SSADM and Object-Oriented based on quality factors to discover the similarities and differences.

Currently, the most widely used methodology for constructing software system is SSADM. This methodology was introduced by Yourdon and Constantine [10], and has since been developed and modified by a number of authors [9, 13, 5, 4, 11]. SSADM methodology focuses on the flow of data through the software processes (functions) which each function is a basic design component.

Today, most information systems are built using Object-Oriented methodology [1, 7]. This methodology allows programmer to build software from software parts called objects. An object contains both the data and functions that operate upon the data. An object can only be accessed via the functions it makes publicly available, so that the details of its implementation are hidden from other objects. By the adoption of Object-Oriented methodology, higher productivity, lower maintenance cost and better quality can be achieved.

Measurement is fundamental to any engineering discipline, and software engineering is, or should be, no exception. Software metrics deals with the measurement of the software product and process by which it is developed. The metrics are used to estimate/predict product costs and schedules and to measure productivity and product quality. So software metrics tell us something about the quality of software. In this paper, we discuss some quality metrics (Reliability, Maintainability, Usability and Flexibility) as quality parameters.

Next section briefly reviews some Distinguishing Characteristics of Object-Oriented methodology like encapsulation, inheritance and abstraction. Section three describes some quality factors that are used for evaluating the mentioned systems. Section four presents an Illustrative Case Study Problem. Section five shows the results of comparing SSADM and Object-Oriented used in an exciting software project and discusses the obtained results.

1.2 The distinguishing characteristics

The Object-Oriented methodology includes a set of mechanisms such as inheritance, encapsulation, and abstraction. Each of these characteristics is discussed briefly in the sections that follow:

1.2.1 Abstraction

Abstraction is the best hope for dramatically the difficulty and cost of software development [7]. At a low level of abstraction, a more procedural orientation is taken. Finally, at the lowest level of abstraction, the solution is stated in a manner that can be directly implemented.

1.2.2 Encapsulation

Encapsulation means that all of this information is packaged under one name and can be reused as one specification or program component. For Object- Oriented systems, encapsulation is encompasses the responsibilities of a class, including its attributes, and operation, and the states of the class as defined by specific attributes values [5].

1.2.3 Inheritance

Inheritance is a mechanism that enables the responsibilities of one object to be propagated to other objects. Inheritance occurs throughout all level of a class hierarchy. Because Inheritance is a pivotal characteristic in many Object Oriented systems, many Object Oriented metrics focus on it [5].

1.3 Software quality factors

Software quality is a complex mix of factors that will vary across different applications and the customers who request them [5]. Software quality can be modeled in different ways and quality as a metric depends on software we are measuring [2]. In this paper, we concentrate on some operational quality factors which are discussed briefly in the sections that follow:

1.3.1 Flexibility

Flexibility is an effort required to modify an operational program. In this paper, flexibility of two implemented systems was investigated with creating a new requirement.

1.3.2 Maintainability

Maintainability is the ease with which a program can be corrected if an error is encountered, adapted if its environment changes, or enhanced if the customer desires a change in requirement. In this paper, maintainability of two implemented systems was investigated with creating code modification

1.3.3 Reliability

Reliability metric as a quality factor attempts to measure and predict the probability of failures during a particular time interval, or the mean time to failure (MTTF). In this paper, reliability of two implemented systems was investigated with creating faulty inputs.

1.3.4 Usability

On early models of quality, McCall's and Boehms quality models [14], usability doesn't play big part. But for end-users usability is perhaps the most important quality metric [2, 8]. For a large number of products, it is believed that usability becomes the final arbiter of quality. Usability metric as a quality factor attempts to measure the effort required to learn, operate, prepare input, and interrupt output of program. In this paper, Usability of two implemented systems was investigated with training users (staffs).

1.4 An illustrative case study problem

As a case study, In order to compare Structured System Analysis and Design Methodology (SSADM) and Object-Oriented methodologies, we decided to examine software system produced by each of the methodologies for a single problem domain. For the problem domain, we take a project of Tehran exchange software system and investigate one part of it. The systems' overall purpose was the" management of monitoring factories and

share investigation”. First, the requirements analysis and system design was conducted using SSADM methodology with DFD diagrams by Easy Case software. Second, to complete the modeling, we performed a simulated Object-Oriented requirements and system design with UML diagrams by Rational Rose software. At last, we implemented the system with Delphi programming language, employing SQL Server database management system.

There is a fundamental difference between data flow and object models. In data flow model, we concentrate on organizing system knowledge around a hierarchy of functions, with data organization playing a secondary role. In an object model, the primary organizing concept is that of an object and its embodied data (attributes) and functions (services). Because of this basic difference, it is natural to ask which the superior methodology for system modeling is.

In this paper, we are attempting the resolution of this question based on some quality factors such as flexibility, reliability, maintainability and usability.

1.5 Comparing the two methodologies

1.5.1 Preliminary results

This section presents quality factors (Reliability, Maintainability, Flexibility and Usability) in terms of the time needed to perform the task. In this section, we first review the results of comparing SSADM and Object-Oriented based on these factors and then discuss about the obtained results.

1.5.1.1 Reliability results

In this paper, Reliability of two implemented systems was investigated with creating faulty inputs. Table 1.1 shows the number of failures detected during a particular time interval in two methodologies.

Table 1.1. Measurement of Reliability in terms of Time (Minutes)

SSADM Methodology	Object-Oriented Methodology
0.10	0.10
0.15	0.15
0.23	0.17
0.27	0.18

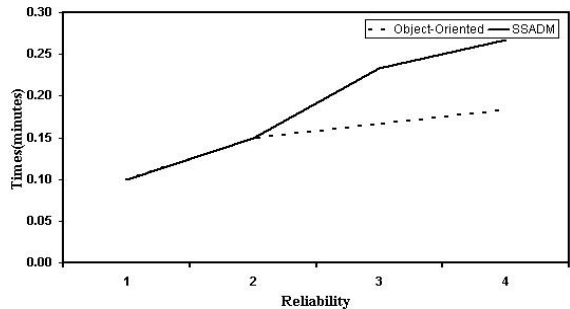


Fig. 1.1. Comparing Reliability of two Methodologies

1.5.1.2 Maintainability results

In this paper, Maintainability of two implemented systems was investigated with creating code modification. Table 1.2 shows the time it takes to correct a change in two methodologies.

Table 1.2. Measurement of Maintainability in terms of Time (Minutes)

SSADM Methodology	Object-Oriented Methodology
37	20
45	30
40	40
85	60

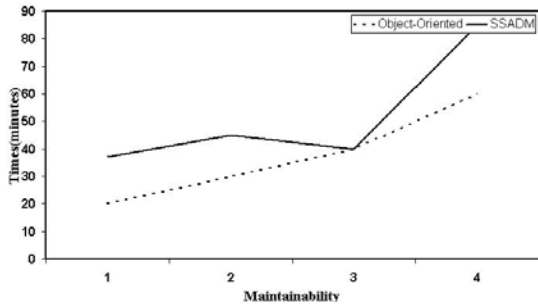


Fig. 1.2. Comparing Maintainability of two Methodologies

1.5.1.3 Flexibility results

In this paper, flexibility of two implemented systems was investigated with creating a new requirement. Table 1.3 shows the time it takes this change in the analysis, design and coding.

Table 1.3. Measurement of Flexibility in terms of Time

SSADM Methodology	Object-Oriented Methodology
60 m	40 m
95 m	55 m
105 m	70 m
185 m	85 m

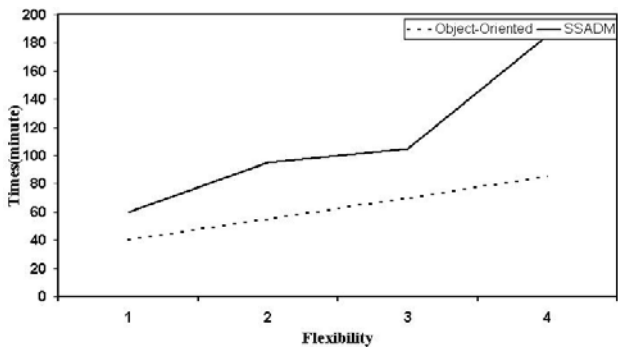


Fig. 1.3. Comparing Flexibility of two Methodologies

1.5.1.4 Usability results

In this paper, Usability of two implemented systems was investigated with training users (staffs). Learnability measured in time needed for learning perform tasks or required training. Thirty-two staffs participated in this study. There were 13 female and 19 male participants ranging in age from 22 to 38. Of the 32 participants, 19 learned SSADM in less time, 9 learned Object–Oriented in less time and 4 learned approximately in equal time.

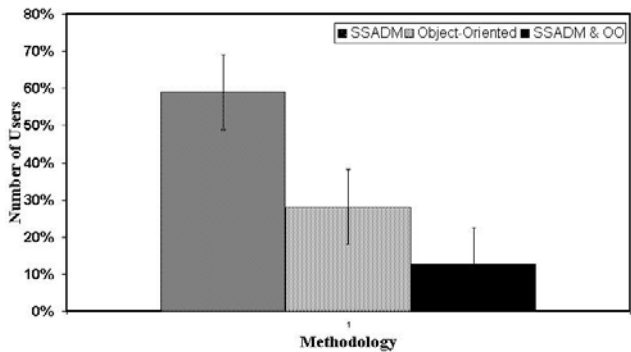


Fig. 1.4. Comparing Usability of two Methodologies

Table 1.4. Measurement of Training in terms of Time

Number of staffs	Methodology
19 (59%)	SSADM
9 (28.12%)	Object-Oriented
4 (12.50%)	SSADM and Object-Oriented

1.5.2 Discussions

This section discusses the results that obtained in the previous section.

1.5.2.1 Differences in reliability

Results showed that the probability of failures during a particular time interval Object-Oriented methodology is fewer than SSADM. The obtained results of this research revealed that Object-Oriented is a suitable methodology based on reliability metric (Figure 1.1).

Theses results are because of some reasons as follow:

- Objects are reusable. Once they are designed, tested and built, objects can be reused in multiple systems and application. Therefore using reusable components can increase reliability.
- Coupling is already proven to be useful criteria for reliability. Interfaces among encapsulated objects are simplified. An object that sends a message need not be concerned with the details of internal data structures. Hence, interfaces is simplified and the system coupling tends to be reduced so reliability increased.
- Encapsulation is the concept that objects should hide their internal contents from other system component and prevents the unauthorized access of an object's data.

1.5.2.2 Differences in maintainability

The obtained results of this research revealed that Object-Oriented is a suitable methodology based on maintainability metric (Figure 1.2).

These results are because of some reasons as follow:

- Object-Oriented methodology consists of independent modules, so independent modules are easier to maintain and test because secondary effects caused by design or code modification are limited and error propagation is reduced.
- Encapsulation can cause the internal implementation details of data and procedures are hidden from the outside world (information hiding). This reduces the propagation of side effects when changing occurs.

- In Object-Oriented methodology, localization is based on objects, and the relationship between operations (functions) and classes is not necessarily one to one. So each object can be changed at any time without affecting the other objects.
- Objects are extensible. They can be changed easily without adversely impacting any previous application that used them.

1.5.2.3 Differences in flexibility

The obtained Results showed that Object-Oriented methodology handle creating a new requirement much faster than SSADM (Figure 1.3).

These results are because of some reasons as follow:

- Reused of a subroutine or function decrease the need to create new code to perform the same task therefore reuse increases flexibility of software.
- In Object-Oriented methodology, better understanding established by the association of data with operations. By maintaining this tight association, maintainability of the software is facilitated by localizing the impact of change to specific parts of the system.
- Objects are extensible. They can be expanded easily without adversely impacting any previous applications that used them.

1.5.2.4 Differences in usability

The obtained Results showed that time needed for learning perform tasks or required training in SSADM is much faster than Object-Oriented. The obtained results of this research revealed that SSADM is a suitable methodology based on Usability metric (Figure 1.4).

These results are because of SSADM does not require very special skills and can easily be taught to the staff. Nowadays common modeling and diagramming tools are used, commercial case tools are also offered in order to be able to set up SSADM easily [3].

1.6 Conclusions

Selecting a suitable methodology for development of a system is difficult. However, some measurements are needed for software engineers in order to choose the suitable methodology. In this paper, we have defined some quality factors such as reliability, maintainability, flexibility and Usability that are practical and easily computable. We introduced two types of methodologies like SSADM and Object-Oriented. In this paper, we compared

two mentioned methodologies based on quality factors to choose the suitable one. The obtained results of this research revealed that Object-Oriented is a suitable methodology based on Reliability, Flexibility and Maintainability, whereas SSADM is an appropriate methodology based on usability factor.

References

1. Cox B (1987) Object Oriented Programming: An evolutionary approach. Addison-Wesley
2. Fenton N, Lawrence S (1997) Software metrics – A rigorous et practical approach. PWS Publishing Company, Revised Printing, pp 337-361
3. ISO (1995) Information Technology- Software lifecycle processes. International Organization for Standardization, ISO/IEC
4. ITS (2003) An introduction to Structured System Analysis and Design Method (SSADM). The Government of Hong Kong Special Administrative Region
5. Jackson M (1983) System development. Prentice-Hall, Englewood Cliffs
6. Jones M (1980) The practical guide to structured systems design. Yourdon Press, New York
7. Korpua, A (2006) Measuring usability. Thesis
8. Meyer B (2003) The power of abstraction, reuse and simplicity: an object-oriented library for event-driven design. Springer-Verlag
9. Nichols D, Twidale B (2002) Usability and open Source Software
10. Pressman R (2002) Adaptable Process Model
11. Rambaugh J, Blaha M, Premerlani W, Eddy F, Lorensen W Object Oriented modeling and design. Prentice Hall
12. Sargent R, Behling P (2002) Back to systems delivery basics with five methodology imperatives.
13. Vazquez F (1994) Selecting a software development process. ACM
14. Yourdon E, Constantine L (1975) Structured design. Yourdon Press, New York

Chapter 2

Sisal 3.2 language structure decomposition

V.N. Kasyanov, A.P. Stasenko

A.P. Ershov Institute of Informatics Systems / Novosibirsk State University,
Novosibirsk, 630090, RUSSIA

Abstract. The functional programming system SFP system being under development at the Institute of Informatics Systems in Novosibirsk is aimed at supporting development of parallel computing applications that still offer high performance and portability. The paper describes equivalent transformations of the Sisal 3.2 programming language (based on Sisal 90) structures. These transformations are to decompose the complex language structures into more simple ones that can be directly expressed by the internal representation IR1, which is based on the intermediate form language IF1. Currently some description of similar transformations can be found in few works about Sisal 90 in the form of examples. These transformations are performed by the front-end compiler from Sisal 3.2 into IR1 and help to better understand its translation strategy. The paper also briefly describes IR1 languages.

Keywords. Compiler, Functional programming, Optimization, Parallel programming, Equivalent transformations

2.1 Introduction

Using traditional methods, it is very difficult to develop high quality, portable software for parallel computers. In particular, parallel software cannot be developed on low cost, sequential computers and then moved to high performance parallel computers without extensive rewriting and debug-

ging. Functional programming [1] is a programming paradigm, which is entirely different from the conventional model: a functional program can be recursively defined as a composition of functions where each function can itself be another composition of functions or a primitive operator (such as arithmetic operators, etc.). The programmer need not be concerned with explicit specification of parallel processes since independent functions are activated by the predecessor functions and the data dependencies of the program. This also means that control can be distributed. Further, no central memory system is inherent to the model since data is not "written" in by any instruction but is "passed from" one function to the next.

Functional language Sisal (Steams and Iterations in a Single Assignment Language) is considered as an alternative to Fortran language for supercomputers [2]. Compared with imperative languages (like Fortran), functional languages, such as Sisal, simplifies programmer's work. He has only to specify a result of calculations and it is a compiler that is responsible for mapping an algorithm to certain calculator architecture (including instructions scheduling, data transfer, execution synchronization, memory management and etc.) In contrast with other functional languages, Sisal supports data types and operators typical for scientific calculations.

At present, there are implementations of the Sisal 1.2 language for many supercomputers, e.g., SGI, Sequent, Encore Multimax, Cray X-MP, Cray 2, etc (see [3]). The Sisal 90 language definition [4] increases the language's utility for scientific programming. It includes language level support for complex values, array and vector operations, higher order functions, rectangular arrays, and an explicit interface to other languages like Fortran and C.

The Sisal 3.2 language that has been designed as the input language of the SFP system being under development at the Institute of Informatics Systems in Novosibirsk [6] is an extension of the Sisal 3.1 [5] which is based on the Sisal 90. Sisal 3.1 simplifies, improves, extends and more exactly specifies Sisal 90. Sisal 3.2 also incorporates ideas of enhanced module support, annotated and preprocessing of Sisal 3.0 [7]. Other extensions of Sisal 3.2 are function overloading and user-defined types, which allow user-defined operations.

The SFP system is aimed at supporting development of parallel computing applications that still offer high performance and portability. It works on a personal computer under Microsoft Windows 2000/XP and provides means to write and debug Sisal-programs regardless target architectures as well as to translate the Sisal-programs into optimized imperative programs, appropriate to the target execution platforms.

The SFP system uses intermediate language R1 [8], which is based on the intermediate form language IF1 [9]. IR1 is a language of hierarchical

graphs [10] made up simple and compound computation nodes, edges and ports. Nodes correspond to computations. Simple nodes are vertices of underlying graph and denote operations such as add or divide. Compound nodes are fragments (or subgraphs) of underlying graph and represent compound constructions such as structured expressions and loops. Ports are vertices of underlying graph that are used for input values and results of compound nodes. Edges show the transmission of data between simple nodes and ports; types are associated with the data transmitted on edges.

In the paper equivalent transformations of the Sisal 3.2 programming language structures are described. These transformations aim to decompose the complex language structures into more simple ones that can be directly expressed by the internal representation IR1. Currently some description of similar transformations can be found in few works about Sisal 90 in the form of examples. These transformations are performed by our front-end compiler and help to better understand its translation strategy. The paper also briefly describes IR1 language.

The rest of the paper is structured as follows. Sections 2.2 and 2.3 briefly describe the most important *Select* and *Forall* compound nodes of the intermediate language IR1 that represent the most Sisal 3.2 language syntax constructions. Sections 2.4 – 2.12 present transformations of complex Sisal 3.2 language structures into more simple ones that can be directly represented by IR1.

2.2 The compound node Select

The compound node *Select* has arbitrary number of input ports and non-zero number of output ports. Let $N \geq 3$ be the number of its graphs. The input ports of all graphs are the same as the input ports of the containing compound node and directly receive values from them. All graphs except the first one have the same output ports as the output ports of the containing compound node. One of these $N-1$ graphs is chosen to supply values of its output ports to the output ports of the containing compound node.

The choice is based on the first graph, which has different semantics compared to its prototype from IF1. The first graph has $N-2$ Boolean output ports (edges that end in these ports have Boolean type), which are sequentially checked until the true value is found on the output port number M . In that case, graph number $M+1$ is chosen. If no *true* value is found, then the last graph is chosen. In the case of the erroneous Boolean value, the error values are placed on the output ports of the containing compound node.

It is clear that the *Select* compound node can directly represent the **if** expression of Sisal 3.2. If the **else** branch is omitted, then the last graph is generated to return error values of the corresponding types.

2.3 The compound node *Forall*

The compound node *Forall* uses the following groups of ports and has four graphs described in Table 2.1. Each port group consists of the same ports for each separate compound node *Forall*.

- The group *C* – the constants, imported to the compound node ports.
- The group *R* – the result values, exported from the compound node ports.
- The group *L* – the dependent loop values.
- The group *L2* – the local loop values.
- The group *OL* – the old dependent loop values.
- The group *D* – the loop range generator values.

Table 2.1. Port groups of the compound node *Forall* and its graphs

Graph Number	Graph Name	Input port groups	Output port groups
	<i>Forall</i>	<i>C</i>	<i>R</i>
1	Initialization	<i>C</i>	<i>L</i>
2	Range generator	<i>C</i>	<i>D</i>
3	Loop body	<i>OL, D, C</i>	<i>L, L2</i>
4	Return statement	<i>OL, L, L2, D, C</i>	<i>R</i>

At the beginning, the output ports of the *initialization* graph are computed and their values are used as the values of the port group *OL* on the first iteration (if any) of the *loop body* graph. The *loop body* graph computes its output ports, for each instance of the port group *D*, generated by the *range generator* graph. The *return statement* graph is computed after each loop iteration and after the last iteration, its output port values are used as the values of the compound node. Before the next iteration, the values from the port group *L* are copied to the ports of the group *OL*.

The *return statement* graph contains the reduction nodes that can only (and only they can) supply values to the output ports of this graph. These reduction nodes directly correspond to one-dimensional reductions of Sisal 3.2 and have an additional memory associated with them between loop iterations.

The *range generator* graph also has the unique *Scatter* nodes that can only (and only they can) supply values to the output ports of this graph.

The *Scatter* node has one input and two output ports. The input port has a type of the array or stream of the type T . The first output port has the type T and the second output port has the integer type. The *Scatter* node sequentially emits a new array or stream element with its index for every loop iteration. If there are several *Scatter* nodes, then they emit new values simultaneously until at least one of them can do it, while the others that cannot, return the error values.

The compound node *Forall* can represent any one-dimensional **for** expression, which is controlled with a range.

2.4 Decomposition of the case expression

The conditional expression **case** is naturally decomposed into the conditional expression **if** with additional **elseif** branches. Since the **if** expression with arbitrary number of **elseif** branches can be already explicitly expressed by the IF1 language, its further simplification is not performed by the Sisal 3.2 front-end compiler and is not shown in this paper.

Every selection list of the **case** expression is transformed into one **if** or **elseif** condition using logical disjunction and conjunction operations over the comparison operation results: equality ($=$), “less than or equal to” ($\leq$) and “greater than or equal to” ($\geq$). For expressions “**case tag**” and “**case type**”, the infix operation **tag** (**tag** function of Sisal 90) and the expression “**type**[...]” are used, respectively.

2.5 Decomposition of the multidimensional loops

Let us consider the following n -ary m -dimensional loop, where each reduction corresponds to one loop dimension (for the simplicity of further notation):

```
for  $D_1$  cross  $D_2$  repeat  $B$ 
  returns  $RN_1$  of  $RV_1$ ; ...;  $RN_n$  of  $RV_n$ 
end for
```

The name D_1 denotes the loop range generator part without the operator **cross** and multidimensional indices of the **at** construction, the name D_2 denotes the remaining part of the range generator, the name RN_i ($i \in 1, \dots, n$) denotes the reduction name with possible initial values and the name RV_i denotes reduction loop values. In that notation m -dimensional loop expres-

sion can be decomposed into the following two loop expressions of dimensions 1 and $m-1$:

```

for  $\overline{D_1}$  repeat
   $\overline{x_1}, \dots, \overline{x_n} :=$  for  $\overline{D_2}$  repeat  $B$ 
    returns  $RN'_1$  of  $RV_1$ ; ...;  $RN'_n$  of  $RV_n$ 
  end for
returns  $RN''_1$  of  $RV_1$ ; ...;  $RN''_n$  of  $RV_n$ 
end for

```

The name $\overline{x_i}$ here and any other name with overline without a special note later denote any unique name. The names RN'_i and RN''_i depend on the name RN_i as shown in Table 2.2.

Table 2.2. Decomposition rules for multi-dimensional reductions, that show how to determine the names RN'_i and RN''_i , which are used in this section before, from the name RN_i

Value of the RN_i .name	RN'_i	RN''_i
Equals to value, product, least, greatest, concatenate, "concatenate(...)" or user-defined reduction.	RN_i	value
Equals to "array[k]($i_1, \dots, i_k$)", where - part "[k]" is optional and equals to "[m]" by default;	$k > 1$ array[k-1]($i_2, \dots, i_k$) $k = 1$	array(i_1) concatenate(i_1)
- last indices of the part "($i_1, \dots, i_k$)" are optional like this whole part and equal to 1 if omitted.	array[1]($i_2, \dots, i_k$)	
Equals to "stream[k]" where part "[k]" is optional and equals to "[m]" by default.	$k > 1$ stream [k-1] $k = 1$ stream [1]	stream concatenate

If the range generator contains multidimensional indices " n in S at $j_1, \dots$ " before the operator **cross**, then the multidimensional loop can be represented in the following way:

```

for  $\overline{D_3}$   $n$  in  $S$  at  $j_1, \overline{D_4}$  repeat  $B$ 
  returns  $RN_1$  of  $RV_1$ ; ...;  $RN_n$  of  $RV_n$ 
end for

```

The name $\overline{D_3}$ denotes the range generator part without the operator **cross** and multidimensional indices of the construction **at**, the name S denotes the array or stream source of multidimensional indices, the name $\overline{D_4}$ denotes the remaining part of the range generator. In that notation m -dimensional loop expression can also be decomposed into the following two loop expressions of dimensions 1 and $m-1$ (where the names RN'_i and RN''_i depend on the name RN_i as shown in Table 2.2):

```

for  $\overline{D_3}$  n in  $\overline{S}$  at  $j_1$ , repeat
   $\overline{x_1}, \dots, \overline{x_n} :=$  for  $n$  in  $\overline{n_1}$  at  $D_4$  repeat  $B$ 
    returns  $RN_1'$  of  $RV_1; \dots; RN_n'$  of  $RV_n$ 
  end for
  returns  $RN_1''$  of  $\overline{x_1}; \dots; RN_n''$  of  $\overline{x_n}$ 
end for

```

2.6 Decomposition of the array element selection

Let us represent the element selection expression from the array A as “ A [*selection construction*]”. If a selection construction does not have the cross (or comma) operator, then it can be represented as “ D_1 **dot** D_2 **dot** ... **dot** D_m ”, where $m \geq 1$ and all expressions $D_1, D_2, \dots, D_m$ are ranges (as required by the operator dot semantics). If $m = 1$ and the part D_1 is a singlet, then the array element selection operation can be represented directly in the IR1 and does not require further decomposition, otherwise the array element selection operation can be decomposed into the following one-dimensional loop:

```

for  $x_1$  in  $D_1$  dot  $x_2$  in  $D_2$  dot ...  $x_m$  in  $D_m$ 
   $\overline{A_1} := A [x_1, x_2, \dots, x_m]$ 
  returns array of  $\overline{A_1}$ 
end for

```

The name x_i (here and later) denotes any unique name, if the part D_i does not have the form “*name* N **in** D_i ”, and denotes the name N otherwise. If the selection construction contains the operator **cross** (or comma), then it can be represented as “ $S_1, S_2, \dots, S_m$ **cross** C_1 ” or “ D_1 **dot** D_2 **dot** ... **dot** D_m **cross** C_2 ”, where $S_1, S_2, \dots, S_m$ denote singlets and the names C_1 and C_2 denote the remaining parts of the selection construction and additionally the part C_1 does not begin with a singlet.

The array element selection operation beginning with a singlet can be decomposed into the following let expression (further decompositions need to be applied recursively):

```

let  $\overline{A_1} := A [S_1, S_2, \dots, S_m]$  in  $\overline{A_1} [C_1]$  end let

```

The array element selection operation beginning with a range can be decomposed into the following one-dimensional loop (further decompositions need to be applied recursively):

for x_1 **in** D_1 **dot** x_2 **in** D_2 **dot** ... x_m **in** D_m **repeat**

$\overline{A}_1 := A[x_1, x_2, \dots, x_m]$

returns array of $\overline{A}_1[C_2]$

end for

The presented decomposition of the array element selection operation also explains the additional restriction, which is missed in Sisal 90 user manual, for the selection construction triplets with omitted parts: they must be placed as the first operand of the selection construction or just after the **cross** operator. In the range D_1 the first and second omitted triplet parts are explicitly represented via “liml(A)” and “limh(A)”, correspondingly. In the ranges $D_2, \dots, D_m$ the triplet parts cannot be omitted because there is no corresponding univocal array dimension available whose lower and upper bounds can be taken. In summary, any array element selection operation was decomposed into the array element selection with simple indices.

2.7 Array element replacement with a singlet list selection construction

This section continues to use the notation of selection construction introduced before. The array element replacement expression in a general form looks like

“A[selection construction := replacement construction]”.

As it shown further, any array element replacement expression can be decomposed into the array one-element replacement with simple index.

If the selection construction is a singlet list $S_1, S_2, \dots, S_n$, then the replacement construction is allowed to be the expression $E_1, \dots, E_t$ and the array element replacement operation is elementary represented as a composition of the following one-element replacement operations:

$A[S_1, S_2, \dots, S_n := E_1][S_1, S_2, \dots, (S_n)+1 := E_2] \dots [S_1, S_2, \dots, (S_n)+(t-1) := E_t]$

2.8 Array element replacement with a scalar replacement construction

Let us consider the case then the selection construction is not a singlet list and the replacement construction is an expression of type of the n -th dimension of the array A , where n is the number of the selection construc-

tion ranges and singlets. In this case, the array element replacement operation can be decomposed into nested one-dimensional loops, obtained after the recursive application of the decompositions given below.

If the selection construction does not have the cross (or comma) operator, the array element replacement operation can be decomposed into the following one-dimensional loop:

```
for  $x_1$  in  $D_1$  dot  $x_2$  in  $D_2$  dot ...  $x_m$  in  $D_m$ 
   $A := \text{old } A [x_1, x_2, \dots, x_m := \text{replacement construction}]$ 
returns value of  $A$ 
end for
```

The array element replacement operation beginning with a singlet can be decomposed into the following **let** expression (further decompositions need to be applied recursively):

```
let  $\overline{A_1} := A [S_1, S_2, \dots, S_m]$ 
  in  $\overline{A_1} [C_1 := \text{replacement construction}]$ 
end let
```

The array element replacement operation beginning with a range can be decomposed into the following one-dimensional loop (further decompositions need to be applied recursively):

```
let  $\overline{A_1} := A$ 
  in for  $x_1$  in  $D_1$  dot  $x_2$  in  $D_2$  dot ...  $x_m$  in  $D_m$ 
     $\overline{A_2} := \text{old } \overline{A_1} [x_1, x_2, \dots, x_m]$ 
     $\overline{A_3} := \overline{A_2} [C_2 := \text{replacement construction}]$ 
     $\overline{A_1} := \text{old } \overline{A_1} [x_1, x_2, \dots, x_m := \overline{A_3}]$ 
    returns value of  $\overline{A_1}$ 
  end for
end let
```

2.9 Array element replacement with array replacement construction

Let us consider the case then the selection construction is not a singlet list and the replacement construction is an expression of type of a k -dimensional array of elements that have the type of the n -th dimension of

the array A . In the considered case, k should be a sum of ranges in the selection construction minus the number of its dot operators. In this case, the array element replacement operation can be also decomposed into nested one-dimensional loops, obtained after the recursive application of decompositions given below.

If the selection construction does not have the cross (or comma) operator, the array element replacement operation can be decomposed into the following one-dimensional loop:

```

let  $\bar{i} := 1$ 
  in for  $x_1$  in  $D_1$  dot  $x_2$  in  $D_2$  dot ...  $x_m$  in  $D_m$ 
     $A := \text{old } A [$ 
       $x_1, x_2, \dots, x_m :=$ 
         $(\text{replacement construction})[\bar{i}]$ 
     $];$ 
     $\bar{i} := \text{old } \bar{i} + 1$ 
  returns value of  $A$ 
end for
end let

```

The array element replacement operation beginning with a singlet can be decomposed into the following let expression (further decompositions need to be applied recursively):

```

let  $\bar{A}_1 := A[S_1, S_2, \dots, S_m]$ 
  in  $\bar{A}_1 [C_1 := \text{replacement construction}]$ 
end let

```

The array element replacement operation beginning with a range can be decomposed into the following one-dimensional loop (further decompositions need to be applied recursively):

```

let  $\bar{A}_1 := A; \bar{i} := 1$ 
  in for  $x_1$  in  $D_1$  dot  $x_2$  in  $D_2$  dot ...  $x_m$  in  $D_m$ 
     $\bar{A}_2 := \text{old } \bar{A}_1 [x_1, x_2, \dots, x_m]$ 
     $\bar{A}_3 := \bar{A}_2 [C_2 := (\text{replacement construction})][\bar{i}];$ 
     $\bar{A}_1 := \text{old } \bar{A}_1 [x_1, x_2, \dots, x_m := \bar{A}_3]$ 
     $\bar{i} := \text{old } \bar{i} + 1$ 
  returns value of  $\bar{A}_1$ 

```

end for
end let

2.10 Decomposition of the where expression

The Sisal 3.2 *where* expression is decomposed into nested one-dimensional loops in the following way, where A , n , R and I are some values

for $\overline{A_1}$ **in** A **returns array of**
 for $\overline{A_2}$ **in** $\overline{A_1}$ **returns array of**
 ...
 for I **in** $\overline{A_{n-1}}$ **returns array of**
 expression R
 end for
 ...
 end for
end for

2.11 Decomposition of the vector operations

All vector operations are decomposed into one-dimensional loops. An operation on multidimensional vectors is decomposed into a vector operation on vectors of lower dimensions.

Prefix and postfix operations on arrays $op(A)$ decompose into:

for $\bar{i}$ **in** A **returns array** ($\text{liml}(A)$) **of** $op(\bar{i})$ **end for**

Prefix and postfix operations on streams $op(S)$ decompose into:

for $\bar{i}$ **in** S **returns stream of** $op(\bar{i})$ **end for**

An infix operation op on two arrays A_1 and A_2 decomposes into:

for $\bar{i}_1$ **in** $\overline{A_1}$ **dot** $\bar{i}_2$ **in** $\overline{A_2}$
 returns array of $\bar{i}_1$ op $\bar{i}_2$
end for

An infix operation op on array A and stream S decomposes into:

for $\bar{i}_a$ **in** A **dot** $\bar{i}_s$ **in** S
 returns stream of $\bar{i}_a \text{ op } \bar{i}_s$
end for

An infix operation op on array A and scalar value V decomposes into:

for $\bar{i}$ **in** A
 returns array ($liml(A)$) **of** $\bar{i} \text{ op } V$
end for

An infix operation op on stream S and scalar value V decomposes into:

for $\bar{i}$ **in** S
 returns stream of $\bar{i} \text{ op } V$
end for

2.12 Conclusion

In the paper the intermediate language IR1 of the functional programming system SFP for supporting supercomputing has been briefly presented. During translation from Sisal 3.2 to the internal representation IR1 some complex Sisal 3.2 language structures need to be reduced to more unified objects of the IR1-language. These transformations have been shown in the terms of Sisal 3.2 by the decomposition of complex language structures into more simple ones that can be directly represented by IR1. The transformations are performed by the front-end compiler and help to better understand its translation strategy. They can be used also as a basis for formal description of semantics of the Sisal 3.2 language. For a general-purpose machine (without any special hardware support for the operations considered in this paper), the described transformations do not introduce unnecessary inefficiency and open additional optimization opportunities.

Acknowledgments. The authors are thankful to all colleagues taking part in the SFP project. The work was partially supported by the Russian Foundation for Basic Research under grant N 07-07-12050.

References

1. Backus J (1978) Can programming be liberated from the von Neumann style? *Commun ACM* 21:613–641
2. Cann D (1992) Retire Fortran? A debate rekindled. *Commun ACM* 35:81–89
3. Gaudiot J–L, DeBoni T, Feo J, Bohm W, Najjar W, Miller P (2001) The Sisal project: real world functional programming. *Lecture Notes in Computer Science* 1808:45–72
4. Feo J, et al (1995) SISAL 90. In: *Proc. High Performance Functional Computing*, Denver, pp 35–47
5. Stasenko AP, Sinyakov AI (2006) Basic means of the Sisa 3.1 language (in Russian). Preprint N 132, A.P. Ershov Institute of Informatics Systems, Novosibirsk
6. Kasyanov VN, Stasenko AP, Gluhankov MP, Dortman PA, Pyjov KA, Sinyakov AI (2006) SFP – An interactive visual environment for supporting of functional programming and supercomputing. *WSEAS Transactions on Computers* 5:2063–2070
7. Kasyanov VN, Biryukova YuV, Evstigneev VA (2001) A functional language SISAL 3.0 (in Russian). In: *Supercomputing support and Internet-oriented technologies*, Novosibirsk, pp 54–67
8. Stasenko AP (2004) Internal representation of functional programming system SISAL 3.0 (in Russian). Preprint N 110, A.P. Ershov Institute of Informatics Systems, Novosibirsk
9. Skedzielewski SK, Glauert J (1985) IF1 – An intermediate form for applicative languages, version 1.0. Tech. Rep. M-170, Lawrence Livermore National Laboratory, Livermore, CA
10. Kasyanov VN, Lisitsyn IA (2000) Hierarchical graph models and visual processing. In: *Proc. of Intern. Conf. on Software: Theory and Practice*, 16th IFIP World Computer Congress, PHEI, Beijing, pp 179–182

Chapter 3

An embedded application for driverless metro train

Vivek Kumar Sehgal,¹ Nitin²

¹Member IEEE and ACM, Department of ECE,

²Member IEEE and ACM, Department of CSE and IT,
Jaypee University of Information Technology, Waknaghat,
Solan-173215, Himachal Pradesh, INDIA
{vivekseh,delnitin}@gmail.com

Abstract. We developed an embedded platform with 8-bit micro-controller 89C51 to implement driverless Metro Train. This sort of work is first time reported in India.

Keywords. Embedded 8-bit processor, Stepper motor, Liquid Crystal Display, Assembly Language, UMPS

3.1 Introduction

Automated Software Controlled Train Simulation (ASCTS) is an example for implementation of a general-purpose rail vehicle application. The trains are equipped with a Central Processing Unit. A human administrator, if required can also access the controls. The train is programmed for a specific path. Each station is well defined i.e. stoppage timing of the train and distance between stations is predefined. ASCTS is built over the Atmel processor 8051, an 8-bit microcontroller with 4KB of on chip ROM, 128 Bytes of RAM, 2 timers and 32 I/O pins. The motion of the train is controlled by a Stepper Motor and for display of messages intelligent Liquid Crystal Display is used.

Since we use software control, the benefit is that the hardware can be standardized i.e. hardware for a new model becomes a question of selecting units from a library of existing units [1]. The design is under research and we propose to add more features to it also. The simulator would have more of user-defined variables. Reports would be generated for multiple trains on a single track [2]. This sort of real time implementation would pave a way to the success of country by reducing time complexity as well as increasing reliability.

The rest of the paper is organized as follows: Section 3.2 describes the working of designed application and hardware used in it with schematic diagram. Section 3.3 describes the hardware implementation for this embedded application. Section 3.4 describes the assembly language code and software for implementation and Section 3.5 describes Challenges followed by conclusion and future scope.

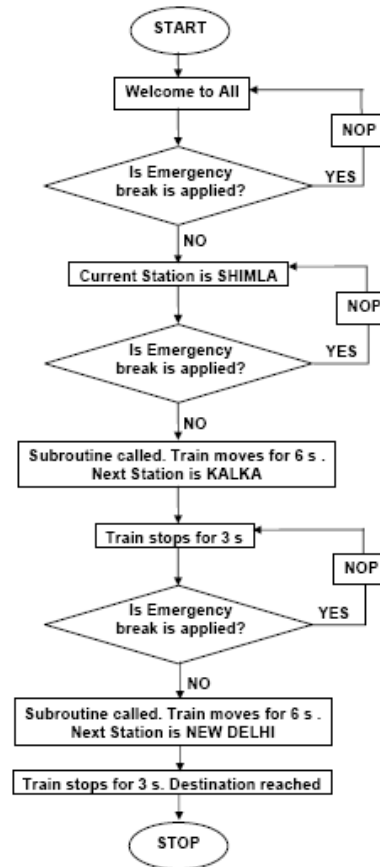


Fig. 3.1. Flow diagram of embedded application

3.2.1 Hardware

The ASCTS hardware consists of

1. Atmel's AT89C51
2. 2X16 Liquid Crystal Display
3. ULN2003
4. A step down transformer: 220:9 for power supply
5. A Buzzer

3.3 Hardware implementation

We implemented the solution in the form of hardware as depicted in the circuit diagram. In the circuit, the LCD display is connected with the P1 of the MC. Control lines are connected with port 3 of the microcontroller. 10K variable resistor controls the contrast of the LCD. A unipolar Stepper motor is used for running of the train. This motor has 5 wires, which are named as A1, B1, A2, B2 and COM. Common line is given at +5V. The other lines can be connected with port 2 of the microcontroller. The ULN 2003 chip derives the stepper motor. This Chip includes Darlington pairs, so that motor can get enough current for its running. This chip requires pull-ups at inputs. Push button is placed at pin no. 12, whose default state is logic 1 & when switch is pressed then logic 0 is applied on the pin. This logic 0 causes 8051 Embedded Processor to be interrupted.

3.3.1 The Resulting Hardware

Here Figure 3.3 shows the embedded circuitry for Metro Train after implementing our design on the board, and Figure 3.4 shows the side view of running model results i.e. the desired hardware model

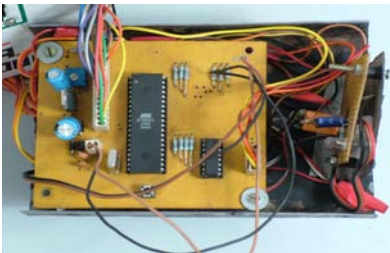


Fig. 3.3. Embedded circuitry for Metro Train

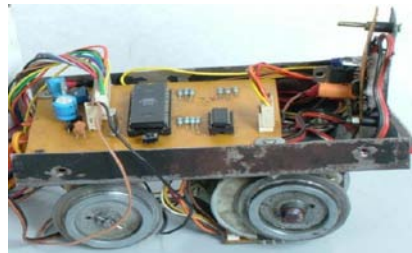


Fig. 3.4. Side view of running model

3.4 Simulations

The system programming is done using assembly language. The program handles things as:

1. Displaying interactive messages on the LCD.
2. Giving the right amount of rotations to the stepper motor at regular intervals.
3. Stopping the train for the required time interval.
4. Stopping the train as soon as the passenger activates the emergency braking system.
5. Controlling the forward as well as the backward motion of the train.
6. Calling the subroutines as is required.

3.4.1 Assembly code (written on UMPS Version 2.1)

The coding of the metro train prototype is given in the assembly language. The program main routines are the routines for running of stepper motor in forward direction as well as in reverse direction. The routines for this purpose are *stepperf* and *stepperb*. The code also explains the various subroutines in detail that are called for correct display of interactive messages on the LCD as well as controlling the overall movement of the train on the desired path.

```

data equ p1
busy equ p1.7
rs equ p3.5
rw equ p3.4
en equ p3.3
bzt equ p0.2
ledf equ p0.0
ledb equ p0.1

org 400h
ajmp main
org 0003h
test: mov c,p3.2
jnc halt
setb bzt
reti
halt:
clr bzt ;till zero blow
on the bzt
ajmp test

main:
mov ie,#00h
setb ea
setb ex0
here:
mov p2,#00h
acall ini
mov dptr,#show0

acall read
clr ledf ;p1.0
acall delay

mov a,#01h
acall command; Now
make memory clear cursor
home
mov dptr,#show1
acall read
setb ex0 #####
mov a,#0c0h
acall command
mov dptr,#show3
acall read
acall delay;;Stopage1
time 3 sec Shimla
loopa1:
acall delay
clr bzt
acall delay

mov a,#01h
acall command
mov dptr,#show2
acall read
mov a,#0c0h
acall command
acall delay;;Stopage2
time 3 sec Kalka

mov a,#01h
acall command
mov dptr,#show2
acall read
mov a,#0c0h
acall command
setb bzt
acall delay10

acall read
setb bzt
acall delay10
acall stepperf

mov a,#01h
acall command
mov dptr,#show1
acall read
mov a,#0c0h
acall command
mov dptr,#show5
acall read
acall delay
;Stopage3 time 3 sec
New Delhi
acall delay
clr bzt
acall delay

setb ledf ; p1.0;off
led at p1.0 for forward journey
clr ledb ; p1.1 ; On
Led for back ward
journey
mov a,#01h
acall command
mov dptr,#show2
;display ne Kalka
acall read
mov a,#0c0h

```

```

acall command
mov dptr,#show4
acall read
setb bzz
acall delay10
acall stepperb
mov a,#01h
acall command
mov dptr,#show1
acall read
mov a,#0c0h
acall command
mov dptr,#show4
acall read
acall delay ;Stopage2 time
3 sec Kalka
acall delay
clr bzz
acall delay
mov a,#01h
acall command
movdptr,#show2 ;display
ne Shimla
acall read
mov a,#0c0h
acall command
mov dptr,#show3
acall read
setb bzz
acall delay10
acall stepperb
mov a,#01h
acall command
mov dptr,#show1
acall read
mov a,#0c0h
acall command
mov dptr,#show3
acall read
setb ledb ;p1.1
ljmp here
;routine for stepper motor
;Delay Routine
delay:
push acc
push 00h

push 01h
push p0
push p1
mov r0,#0eh
loopr:
mov a,#0ffh
loopb:
mov b,#0ffh
loopa: djnz b,loopa
djnz 0e0h,loopb
djnz r0,loopr
pop p1
pop p0
pop 01h
pop 00h
pop acc
ret
;Delay stepper
delays:
push acc
push 00h
push 01h
push p0
push p1
mov a,#0ffh****
loopa1:
mov b,#0fh
loopb1:
djnz b,loopb1
djnz 0e0h,loopa1
pop p1
pop p0
pop 01h
pop 00h
pop acc
ret
delay10:
mov tmod,#01h
mov tcon,#00h
mov tl0,#0f0h
mov th0,#0f8h
setb tr0
no: jnb tf0,no
clr tr0
clr tf0
ret

Routine to read data from
prog mem
read:
nex: clr a
movc a,@a+dptr
cjne a,#'0',aga
sjmp down
aga: acall display
sjmp nex
down:
ret
stepper routine
stepperf:
push acc
push p1
mov a,#88h
mov r1,#04h
loop1:
mov r0,#0e0h
loop: mov p2,a
acall delays
rr a
djnz r0,loop
djnz r1,loop1
pop p1
pop acc
ret
stepperb:
push acc
push p1
mov a,#88h
mov r1,#04h
loop12:
mov r0,#0e0h
loop0: mov p2,a
acall delays
;LCD strobe subroutines
ini:
mov a,#38h
acall command
mov a,#38h
acall command
mov a,#38h
acall command
mov a,#38h
acall command
mov a,#38h
acall command

mov a,#0eh
acall command
mov a,#06h
acall command
mov a,#01h
acall command
mov a,#80h
acall command
ret
command:
acall ready
mov data,a
clr rs
clr rw
setb en
clr en
ret
display:
acall ready
mov data,a
setb rs
clr rw
setb en
clr en
ret
ready:
clr en
mov data,#0ffh
clr rs
setb rw
wait:
clr en
setb en
jb busy,wait
clr en
ret
end

```

3.5 Challenges

The domain of railways represents a huge variety of possibilities for applications of software. The realm of automatic control interacts strongly with many facets of statistics gathering, planning, train scheduling, resource allocation etc [3]. The field of computing science provides design tools, techniques, principles and methods that allow classical automatic control engineering designs to link up to more modern operations analysis, graph theoretic and combinatorial resource optimization [4].

3.6 Conclusion and future scope of this model

We developed an embedded platform with 8-bit microcontroller 89C51 to implement driverless metro train. This sort of work is first of its kind in India. This sort of system could make the railways network more efficient and reliable.

Optimum throughput can be achieved through the mapping of tracks. Another feature of the system could be implementation of vital remote controls on software. We need to define safe failures also in the control system. A safe failure occurs when the failure of a component in a processing system is detected and the effect of the failure emerges as a safe-side operation [4]. In the railway application, the safe-side failures are defined for the outputs of the controller. Failures in wayside control functions can be safely handled by shutting down the microprocessor-based controller. In addition, Adhoc methodologies to safety-critical design can be implemented in these systems to ensure safety and fault-tolerant operation.

Acknowledgement. This model got the Third Prize at The Annual Techfest – TROIKA 2007, Sponsored by IEEE and IEE Student Chapters of Delhi College of Engineering at DCE, New Delhi in INDIA.

References

1. Astrom P (1992) Control electronics on rail vehicles. In: Proceedings of the ASME/IEEE Spring Joint Railroad Conference, pp 107-116
2. Gordon SP, Leherer DG (1998) Coordinated train control and energy management control strategies. In: Proceedings of the 1998 ASME/IEEE Joint Railroad Conference, Philadelphia, PA, USA, pp 165-176
3. Raymond Abrial Jean (2000) Formal software techniques in Railway Systems. In: Proc. 9th IFAC Symposium on Control in Transportation Systems, Braunschweig, Germany, pp 1-12
4. Hachiga A (1993) The concepts and technologies of dependable and real time computer systems for Shinkansen train control. Dependable computing and fault tolerant systems, Springer Verlag, Berlin, pp 225-252
5. Mazidi MA, Mazidi JG (2006) 8051 microcontroller and embedded system. Prentice Hall

Chapter 4

A post-mortem JavaSpaces debugger

Jason Howarth, Edward Stow

School of Computing and Mathematics, Charles Sturt University, Boorooma St. Wagga Wagga, NSW Australia 2678

Abstract. We describe a post-mortem debugger called JavaSpaces Debugger (JSD) that is able to detect a class of properties that occur in a distributed JavaSpaces program. To detect a property using JSD, the user specifies a global predicate to be evaluated. This global predicate is divided into a series of local predicates that are evaluated by each system node. During execution, an event message is sent to the debugger whenever one of these local predicates is true. The debugger arranges event messages according to logical time, and reports whether the global predicate occurred. JSD can detect both weak and strong unstable predicates, using integer, boolean, or bytecode expressions.

Keywords. JavaSpaces, Jini, distributed debugging, strong unstable predicates, weak unstable predicates, vector clock, Java Virtual Machine, Java Virtual Machine Debugging Interface

4.1 Introduction

Distributed computing systems are characterised by the absence of a global time base, unpredictable message delays, and the dispersion of program state amongst nodes [10]. This makes it difficult to capture and analyse the state of such systems.

We describe JSD (JavaSpaces Debugger), a debugger able to overcome some of these limitations. JSD is used to detect a class of properties that occur in a distributed JavaSpaces program.

To detect a property using JSD, the user first specifies a global predicate to be evaluated. This global predicate is divided into a series of local predicates that are evaluated by each system node. During execution, an event message is sent to a global checker process whenever one of the local predicates is true. The checker process arranges these messages according to logical time, and reports whether the global predicate occurred.

JSD is used to detect global predicates classified as unstable. An unstable predicate is one that may alternate between true and false during execution. Unstable predicates may be either weak or strong [5, 6].

A strong unstable predicate, where true, defines a property that certainly occurred in an observed run of the monitored program. A weak unstable predicate, where true, specifies a property that may have occurred, but which cannot be proven. JSD can detect both strong and weak unstable predicates.

4.2 Background

The following sections contain a brief background on the technologies and algorithms used in our debugger.

4.2.1 JavaSpaces

JavaSpaces is based on the shared-memory model of distributed computing [1]. Instead of cooperating directly, processes exchange Java objects through a network storage area, called a JavaSpace. This makes JavaSpaces different to the message-exchanging framework common to many distributed systems. JavaSpaces has been used for cluster computing [2], software agent systems [11] and distributed programming generally [3].

The main operations performed on a JavaSpace are *write*, *read*, *take*, and *notify*. The *write* method deposits an object in the JavaSpace. To *read* an entry, it is necessary to supply a template object to match against. If a match is found, a copy of the matched object is returned from the JavaSpace. The *take* method acts the same as *read*, except the entry is deleted from the JavaSpace. The *notify* method allows a process to register interest in future entries that appear.

Although the JavaSpaces API (Application Programmer's Interface) is minimal, complex distributed algorithms can be constructed via the flow of objects in and out of JavaSpaces [4].

4.2.2 Jini

JavaSpaces rests upon the Jini foundation, a technology used to create networked communities of software and hardware services. JavaSpaces is provided as a standard Jini service. Like all Jini services, JavaSpaces is used through its service proxy, which is stored at a Jini Lookup Service. The Lookup Service is a registry of all currently active services available within a Jini community.

To retrieve a service proxy, the lookup repository is searched for services matching certain criteria. For example, to retrieve the JavaSpaces proxy, a client might search the Lookup Service for a proxy with a name attribute of 'JavaSpaces'. Once the client has the proxy, it can make local method calls on this object to access the backend service.

JavaSpaces plays an important role in the overall Jini architecture by providing a network storage facility. This allows Jini clients to store objects centrally, as well as allowing clients to communicate via the flow of objects in and out of the JavaSpace.

4.2.3 Weak Unstable Predicates

We turn to an approach for detecting weak unstable predicates. Such predicates may be either conjunctive or disjunctive. A conjunctive predicate is of the form $s(i) \wedge s(j) \wedge s(k)$, where $s(i)$ represents the local state of process i , $s(j)$ the local state of process j , etc. This paper will focus only on conjunctive predicates. Disjunctive predicates are trivial to detect since only local state must be monitored [5].

The algorithm [5] relies on each process checking its local portion of the predicate. For example, if the predicate is $x=1 \wedge y=2$ for two variables x and y held at processes P_1 and P_2 respectively, then process P_1 monitors x , and process P_2 monitors y . When a local predicate becomes true, the current vector clock timestamp is sent by the local process to a global checker process. Once computation has ended, the checker process evaluates the global predicate to see whether both local predicates could have occurred concurrently. The weak conjunctive predicate $a \wedge b$ is true if a and b occur concurrently in the program trace [5].

The implementation of this algorithm requires a special type of vector clock, called an lcmvector (last causal message vector) [5]. An lcmvector

consists of a vector of n integers, one for each process. Each process is responsible for maintaining its own vector. For a system with three processes, the lcmvector stored at process P_i takes the form $[C_i, C_j, C_k]$, where C_i is the clock that measures progress at P_i , and C_j and C_k are set to values advised by processes P_j and P_k , respectively.

The operations permitted on an lcmvector are initialisation $\langle \text{init} \rangle$, increment $\langle \text{incr} \rangle$, and synchronization $\langle \text{synch} \rangle$. Initially, the values in each vector are set to 0.

The $\langle \text{init} \rangle$ operation occurs at process startup, where each process P_i sets its clock value to $i = 1$. Whenever a message is sent, the $\langle \text{incr} \rangle$ operation is performed. For a process P_i , this involves incrementing its clock by one ($C_i = C_i + 1$). When a message is received, the synchronize operation is invoked. This means that the receiving process performs the componentwise maximum operation, using as input its own clock and the clock of the process sending the message. To implement the algorithm, lcmvectors are passed between processes when a message is sent.

Operations on lcmvectors are depicted as occurring between states. Process events have the clock value that appears before the event. Hence in Figure 4.1, events a and b both have the vector time stamp $\{1, 0\}$.

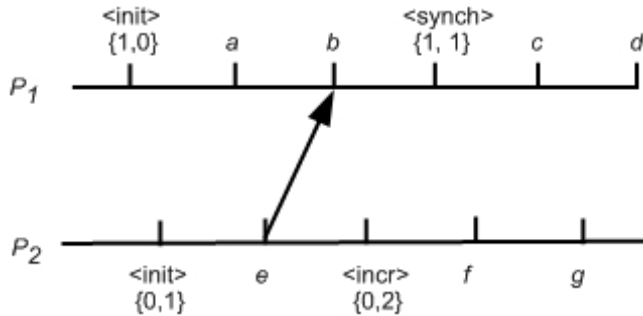


Fig. 4.1. Events measured using lcmvectors

Vector clocks are compared using a type of vector clock algebra, based on the *happened before* [8] relation ($\Rightarrow$). Vector clocks are considered concurrent ($\Leftrightarrow$) if the happened before relation cannot be established.

Consider two clocks V_1 and V_2 . Based on the values contained in these clocks, the following holds true:

$V_1 \Rightarrow V_2$ if $V_1 \neq V_2$ and if each value in V_1 is less than or equal to each value in V_2

$V_2 \Rightarrow V_1$ if $V_1 \neq V_2$ and if each value in V_1 is greater than or equal to each value in V_2

$V_1 \Leftrightarrow V_2$ if neither $V_1 \Rightarrow V_2$ nor $V_2 \Rightarrow V_1$

Applying these rules to the example in Figure 4.1, we can assert that $e \Rightarrow c$, and that $c \Leftrightarrow f$.

We turn to the checker process to see how the global predicate is evaluated. The checker process enqueues clock values received as `<debug>` messages (Figure 4.2) whenever a portion of the global predicate is detected locally. For example, if the global predicate is $x = 4 \wedge y = 3$, process P_1 sends a message to checker process C whenever its portion of the predicate is true (see Figure 4.2); process P_2 does likewise.

The messages sent to the checker process C contain the timestamps associated with detection of the local predicate. The checker process uses the vector clock comparison rules to decide if the predicate is true. In the case of a weak unstable predicate, the predicate $i \wedge j$ is true if $i \Leftrightarrow j$ [5].

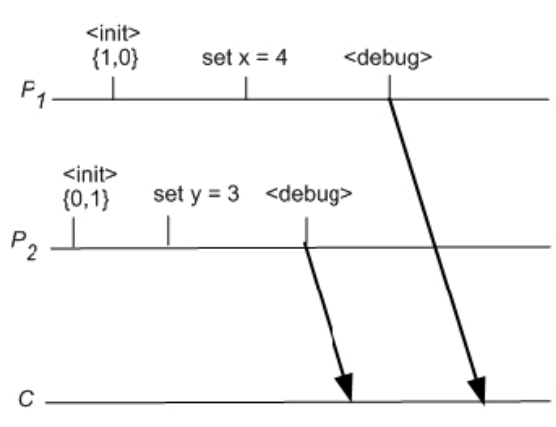


Fig. 4.2. Time-stamped debug messages sent to the checker process

4.2.4 Strong unstable predicates

Strong conjunctive predicates are of the form $s(i) \wedge s(j) \wedge s(k)$, where $s(i)$ represents the local state of process i , $s(j)$ the local state of process j , and so on. In detecting such predicates, we rely on the notion of *intervals* [6]. An interval is a pair of vector clocks that delineate a period during which the predicate was true. In the following discussion, *clock_lo* refers to the start, and *clock_hi* to the end of an interval.

Consider two processes with a boolean variable *ready_to_commit*. Assume it must be determined if there is a point when these processes on Hosts A and B are simultaneously ready to commit. To achieve this, it must be determined if the intervals where the predicate is detected on each host overlap.

For example, assume host *A* is ready to commit during the interval $\{2, 1\}$ to $\{4, 2\}$, and host *B* during the interval $\{0, 1\}$ to $\{2, 2\}$. If these intervals overlap, then there must have been a time when the global predicate (i.e., when both hosts were simultaneously ready to commit) is also true.

To determine overlap, the rules for comparing vector clocks are applied to see if $\text{host } A.\text{clock\_lo} \Rightarrow \text{host } B.\text{clock\_hi}$ AND $\text{host } B.\text{clock\_lo} \Rightarrow \text{host } A.\text{clock\_hi}$. If so, these intervals have overlapped in (logical) time and the global predicate must have occurred [6]. In the example above, since $\{2, 1\} \Rightarrow \{2, 2\}$ and $\{0, 1\} \Rightarrow \{4, 2\}$, these intervals overlap, and it is certain that the global predicate was true at some point in the computation.

4.3 JSD architecture

A feature of JSD is the level of indirection added to the JavaSpaces architecture to capture debugging information. A typical JavaSpaces client interacts with the JavaSpace using the JavaSpaces proxy. In JSD, the debugged host is modified to load a different proxy, called the *debugging proxy*. The debugging proxy does not communicate directly with the JavaSpace; instead, it routes messages through another proxy-like object, *JSDDebugSpace*.

JSDDebugSpace is a custom-written Jini service used to filter messages intended for the real JavaSpace. Information is captured about these messages before they are passed to the JavaSpace for normal processing. Similarly, when a message is returned from the JavaSpace, it returns through *JSDDebugSpace* before it is handed back to the debugged application. By this means, debugging control is interposed between the debugged program and the JavaSpace.

Both *JSDDebugSpace* and the debugging proxy on each host maintain a lcmvector clock to order system events. These clocks are exchanged between the proxy and *JSDDebugSpace* whenever a JavaSpaces call is made. By this means, clocks in the debugged system are kept current through the increment and synchronization methods described earlier.

To monitor the local predicate, the debugging proxy loads a DLL (dynamic link library) on the host being debugged. This DLL uses the JVMDI (Java Virtual Machine Debugger Interface) [7]. The DLL is passed details of the predicate to watch. Based on the contents of this predicate, it then registers for *watchpoint* events with the local JVM. When such an event occurs, the JVM notifies the DLL, which in turn notifies a checker process with the lcmvector reading at the local host. These readings are put in a queue maintained by the checker process. The contents of this queue are

later used to reconstruct the event sequence that occurred during monitoring. Once the debugged program ends, the truth of the global predicate first entered by the user can be determined.

4.3.1 Predicate detection language

We have designed a simple language called JPL (JavaSpaces Predicate Language) to submit predicates to our debugger. JPL is used to express a predicate containing either integer or boolean values. A bytecode index can also be specified. An integer or boolean predicate takes the form *host-name: class name: variable operator result* [$\wedge$]. The $\wedge$ symbol is the conjunctive operator, used to chain local predicates into a single global predicate.

To give an example, the following predicate seeks to determine if variables x and y , at different hosts in the system, are ever equal to 10 at the same point in the computation: *Host A: Adder: $x = 10$ $\wedge$ Host B: Subtractor: $y = 10$* . The local predicates in this example are *Host A: Adder: $x = 10$* and *Host B: Subtractor: $y = 10$* . The variables x and y are declared in classes *Adder* and *Subtractor*, respectively.

JPL also allows predicates that contain more complex expressions. All of the basic mathematical operators are supported. In addition, a boolean predicate can use greater-than or less-than logic, such as *Adder: $i > 5$ $\wedge$ Subtractor: $j < 3$* .

The other type of predicate is the bytecode predicate, which has the structure *hostname: break: class name: method name: bytecode index* [$\wedge$]. An example is *Host A: break: Adder: increment: 44 $\wedge$ Host B: break: Subtractor: decrement: 23*. This is asking: when class *Adder* hits bytecode index 44 in the *increment* method, is it true that class *Subtractor* hits bytecode index 23 simultaneously in the *decrement* method? We believe bytecode predicates offer an advantage over integer and boolean predicates because a variable need not exist in the application being debugged to perform predicate evaluation.

4.4 JSD example

The example we describe is based on a simple chat program that consists of a Listener and two Talkers. To send a message to the Listener, a Talker must first hold an exclusive token T . This token is exchanged between each Talker via a JavaSpace. The Listener is notified through the JavaSpaces *notify* method whenever a message is written to space. Our concern

here is with the token, which should not simultaneously be held by more than one Talker. The predicate for this is as follows: *Host A: Talker: hasToken = true* $\wedge$ *Host B: Talker: hasToken = true*.

If this predicate is true, there is a problem with our algorithm because there is a chance that both hosts can talk at once. The setup is as follows. Hosts Sun and Moon each run a Talker process (Figure 4.3). In this test, Sun.Talker was started first, followed by Moon.Talker. We assume that the token has already been written to the JavaSpace, and contains the name of the host who has first possession. This host takes the token, sends a message to the Listener (not shown), then releases the token again, addressed to the next recipient who performs the same set of actions.

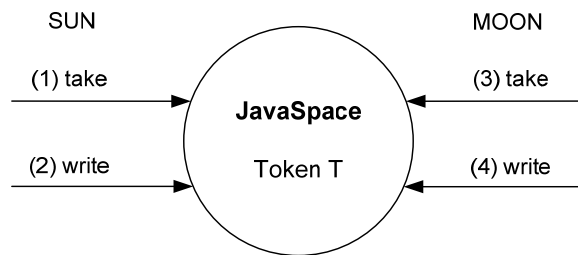


Fig. 4.3. Hosts Sun and Moon take and write the token

After running the test, the following values appear in the local predicates screen of JSD:

Sun

Talker:hasToken = true {2,0,1}

Moon

Talker:hasToken = true {2,2,3}

This shows that each local predicate was true when the lcmvector readings were {2, 0, 1} and {2, 2, 3}, respectively. The three readings in each lcmvector indicate that three clocks were used in this debugging session, one for each host and one for the JavaSpace.

We are now in a position to determine whether the weak predicate was true during this run of the program. To do this, we must determine if it were possible for readings {2, 0, 1} and {2, 2, 3} to have occurred concurrently. When we compare these clocks, we find that $\{2, 0, 1\} \Rightarrow \{2, 2, 3\}$, which means that the weak conjunctive predicate is false, at least under the scenario where host Sun is started first.

In this example, the source code on both hosts contains the boolean variable *hasToken = true* for use in detecting the predicate. If this variable did not exist the predicate could not be assessed. Unfortunately, many dis-

tributed debuggers [3, 9] assume such a variable. JSD overcomes this limitation by allowing the user to specify a bytecode index instead of a variable when constructing the predicate.

We next apply a strong conjunctive predicate to the same scenario - this time to see if it was definitely the case that, at some point in the computation, Sun had the token and Moon did not. The predicate to express this is *Sun: Talker: hasToken = true* $\wedge$ *Moon: Talker: hasToken = false*. Only one round of token passing is used, and Moon is started before Sun. Once the test is run, JSD reports the following intervals:

Sun

Talker: hasToken = true [{2,1,1} {3,1,2}]

Moon

Talker:hasToken = false [{0,1,0} {2,2,3}]

JSD reports that the strong conjunctive predicate is true because intervals $\{[2, 1, 1] [3, 1, 2]\}$ and $\{[0, 1, 0] [2, 2, 3]\}$ overlap. In general, however, due to the loosely-coupled nature of JavaSpaces, a strong conjunctive predicate will never be true unless the processes named in the predicate are both active in the same JavaSpace at the same (physical) time. A time overlap only occurs when both processes are engaged in a common task – that is, when both simultaneously interact with the JavaSpace. This makes strong conjunctive predicates much less useful than weak conjunctive predicates in a JavaSpaces environment.

4.5 Conclusion

We have described JSD, a debugger able to detect both weak and strong unstable predicates in a JavaSpaces program, using the algorithms of [5, 6]. *MaX* [9], a distributed debugger framework for a Java/CORBA environment, also uses these algorithms. There are several important differences between *MaX* and our debugger, including the ability of JSD to detect bytecode predicates. Our design also offers fewer overheads than *MaX* because only one process per machine is needed to run both the original program and the predicate detection algorithms. Hence, our debugger is in process, whereas *MaX* is out-of-process.

The JavaSpaces debugger described in [3] is used to capture message activity between the proxy and the JavaSpace. We believe our approach is superior because, in addition to capturing message activity, JSD can arrange these messages into a global sequence, as well as detect strong and weak conjunctive predicates.

Many enhancements could be made to JSD. Debugging data could be saved in XML format to compare debugging sessions. This would offer a more complete picture than is currently provided. Any predicate judged against this additional data would have more validity than one judged against a single program run. The architecture might also be adapted to an RMI or CORBA environment so that debugging on each host is directed by stub objects, rather than by a Jini proxy.

Acknowledgement. The authors gratefully acknowledge the valuable comments provided on an early draft of this paper by Dr. Barney Dalgarno of Charles Sturt University, Wagga Wagga.

References

1. Attiya H (1998) Distributed computing theory. In: Zomaya A (ed) Parallel and distributed computing handbook. McGraw-Hill, New York
2. Batheja J, Parashar M (2001) Adaptive cluster computing using JavaSpaces. Research Report, Dept. of Electrical and Computer Engineering, Rutgers University, New Jersey
3. Bishop P, Warren N (2003) Javaspaces in practice. Addison-Wesley, London
4. Freeman E, Hupfer S, Arnold K (1999) JavaSpaces: Principles, patterns and practice. Addison-Wesley, Reading, Massachusetts
5. Garg V, Waldecker B (1994) Detection of weak unstable predicates in distributed programs. IEEE Transactions on Parallel and Distributed Systems 5(3):299-307
6. Garg V, Waldecker B (1996) Detection of strong unstable predicates in distributed programs. IEEE Transactions on Parallel and Distributed Systems 7(12):1323-1333
7. Java Platform Debugger Architecture, Sun Microsystems Inc (2002) last viewed May 10, 2007 <<http://java.sun.com/j2se/1.4.2/docs/guide/jpda/>>
8. Lamport L (1978) Time, clocks and the ordering of events in a distributed system. Communications of the ACM 21(7):558-65
9. Otta M (2001) A distributed debugger framework applicable in a Java/CORBA environment. European Research Seminar on Advances in Distributed Systems, last viewed May 1, 2007 <<http://www.cs.unibo.it/ersads/program.html>>
10. Stoller S (1997) Detecting global predicates in distributed systems with clocks. In: Proceedings of the 11th International Workshop on Distributed Algorithms (WDAG97), Springer-Verlag, London, pp 185-199
11. Wang A (2002) Using JavaSpaces to implement a mobile multi-agent system. Research Report, Norwegian University of Science and Technology, Trondheim

Chapter 5

Stochastic numerical simulations using C#

Edita Kolářová,¹ Csaba Török²

¹Brno University of Technology, Technická 8, 616 00 Brno, Czech Republic

²Technical University Košice, Letná 9, 042 00 Košice, Slovakia

Abstract. This paper considers numerical solutions of stochastic differential equations. The stochastic Euler scheme and the stochastic Milstein scheme are examined. We present two different approaches to implementation of these schemes in the MS .NET Framework. The first one is based on matrices and the second one on delegates. The schemes and the codes are illustrated with an example.

Keywords. Stochastic differential equation, Euler scheme, Milstein scheme, C#

5.1 Introduction

The effects of intrinsic noise within physical phenomena are ignored when using deterministic differential equations for their modelling. Stochastic differential equations (SDEs) include a random term which describes the randomness of the system as well. A general scalar SDE has the form

$$dX_t = F(t, X_t) dt + G(t, X_t) dW_t,$$

where $F : \langle 0, T \rangle \times R \rightarrow R$ is the drift coefficient and $G : \langle 0, T \rangle \times R \rightarrow R$ is the diffusion coefficient. W_t is the so called Wiener process (see [2]), a

stochastic process representing the noise. We can represent the SDE in the integral form

$$X_t = X_0 + \int_{t_0}^t F(s, X_s) ds + \int_{t_0}^t G(s, X_s) dW_s,$$

where the first integral is an ordinary Riemann integral. Since the sample paths of a Wiener process do not have bounded variation on any time interval, the second integral cannot be a Riemann-Stieltjes integral. K. Itô proposed a way to overcome this difficulty with the definition of a new type of integral, a stochastic integral which is now called the Itô integral (see [2]). Later R. L. Stratonovich proposed another kind of stochastic integral, called the Stratonovich integral.

5.2 Numerical solutions of SDEs

There are two main approaches to numerical solutions. The first one is based on numerical methods for ordinary differential equations (see [5]). The second one uses more information about the Wiener process (see [1]). We give here the formulae for the simplest examples of these two approaches, the Euler and the Milstein methods.

A general N -dimensional Itô SDE with an M -dimensional Wiener process $\mathbf{W}_t = (W_t^1, \dots, W_t^M)$ can be written componentwise as

$$dX_t^i = F^i(t, \mathbf{X}_t) dt + \sum_{j=1}^M G^{i,j}(t, \mathbf{X}_t) dW_t^j,$$

$i = 1, \dots, N$, where $\mathbf{X}_t = (X_t^1, \dots, X_t^N)$. Let the stochastic process $\mathbf{X}_t, t_0 \leq t \leq T$ be the solution of the multidimensional SDE on $t_0 \leq t \leq T$ with the initial value $\mathbf{X}_{t_0} = \mathbf{X}_0$. Let us consider an equidistant discretisation

of the time interval $t_n = t_0 + nh$, where $h = \frac{T-t_0}{n} = t_{n+1} - t_n$ and the corresponding discretisation of Wiener processes

$$\Delta W_n^j = W_{t_{n+1}}^j - W_{t_n}^j = \int_{t_n}^{t_{n+1}} dW_s^j, j = 1, \dots, M. \text{ To be able to apply any}$$

stochastic numerical scheme, first we have to generate the random increments of the Wiener processes as independent Gauss random variables with $E[\Delta W_n^j] = 0$ and $E[(\Delta W_n^j)^2] = h$.

The Euler scheme for the i -th component of the multidimensional SDE has the form

$$X_{n+1}^i = X_n^i + F^i(t_n, \mathbf{X}_n)h + \sum_{j=1}^M G^{i,j}(t_n, \mathbf{X}_n)\Delta W_n^j.$$

The Milstein scheme for the i -th component of the multidimensional SDE has the form

$$X_{n+1}^i = X_n^i + F^i h + \sum_{j=1}^M G^{i,j} \Delta W_n^j + \sum_{j_1, j_2=1}^M L^{j_1} G^{i, j_2} \int_{t_n}^{t_{n+1}} \int_{t_n}^t dW_s^{j_1} dW_t^{j_2}$$

for $i = 1, \dots, N$, where the operators L^j are defined by $L^j = \sum_{i=1}^N G^{i,j} \frac{\partial}{\partial x_i}$.

The Euler scheme converges with strong order $\gamma = \frac{1}{2}$ and the stochastic Milstein scheme converges with strong order $\gamma = 1$ (see [1]).

5.3 Numerical experiments using C#

In this section we present two different approaches to implementation of the Euler and Milstein methods for numerical solution of SDEs in the MS .NET Framework. The first one is based on matrices and the second one on delegates. To make matrix manipulation and visualization simpler we used the component library LinAlg described in [3]. LinAlg is a set of classes that enables vectorial programming and incorporates a wide range of numerical, statistical and graphical methods.

Matrix arguments. The following codelines present the implementation in MS Visual C# of the Euler and Milstein methods described in section 5.2 for the Itô SDE of form $d\mathbf{X}_t = \mathbf{F}\mathbf{X}_t dt + \mathbf{G}\mathbf{X}_t dW_t$, where X_t is an r dimensional vector, F and G are $r \times r$ matrices and W_t is a one dimensional Wiener process:

```
static public MatrixD Euler(VectorD X0, MatrixD F, MatrixD G, double h, int n, VectorD w)
{
    MatrixD xM = new MatrixD(X0.Length, n);
    VectorD xMj = new VectorD(X0);
    xM.ColumnsSet(0, xMj); double dw;
    for(int j=1; j<n; j++)
    {
        dw = w[j] - w[j-1];
        xMj += F*xMj*h + G*xMj*dw;
    }
}
```

```

        xM.ColumnsSet(j, xMj);
    }    return xM;
}

static public MatrixD Milstein(VectorD X0, MatrixD F,
MatrixD G, MatrixD DG, double h, int n, VectorD w)
{
    MatrixD xM = new MatrixD(X0.Length, n);
    VectorD xMj = new VectorD(X0);
    xM.ColumnsSet(0, xMj);
    double dw;
    for(int j=1; j<n; j++)
    {
        dw = w[j] - w[j-1];
        xMj += F*xMj*h + G*xMj*dw + 0.5*DG*G*xMj*(dw*dw-h);
        xM.ColumnsSet(j, xMj);
    }
    return xM;
}

```

where the approximant $\mathbf{xM}$ of $\mathbf{X}_t$ is a matrix and its k -th column contains the approximation of $\mathbf{X}_t$ in t_k .

Delegate arguments. The C# methods, Euler(...) and Milstein(...), presented above, use matrices as arguments to solve the Itô SDE $d\mathbf{X}_t = \mathbf{F}\mathbf{X}_t dt + \mathbf{G}\mathbf{X}_t dW_t$. Fortunately the MS .NET Framework enables, due to the concept of delegate, the implementation of methods with function arguments that solve the more general equation $d\mathbf{X}_t = \mathbf{F}(t, \mathbf{X}_t) dt + \mathbf{G}(t, \mathbf{X}_t) dW_t$. Delegates are type-safe function pointers that can be used to program events, multithreads, asynchron operations or remoting (see [4]). Now we describe shortly in four steps the implementation of the Euler scheme with function arguments (the steps for the Milstein scheme are similar).

Step 1: declare a public delegate FuncVec with a given signature (double and VectorD arguments) and VectorD return type:

```
public delegate VectorD FuncVec(double t, VectorD X);
```

Step 2: implement a method EulerDelegate with arguments of FuncVec delegate type:

```

static public MatrixD EulerDelegate(VectorD X0, FuncVec
F, FuncVec G, double h, int n, VectorD w)
{
    MatrixD xM = new MatrixD(X0.Length, n);
    VectorD xMj = new VectorD(X0);
    xM.ColumnsSet(0, xMj);
    double dw;
    for(int j=1; j<n; j++)
    {
        dw = w[j] - w[j-1];

```

```

        xMj += F(0, xMj)*h + G(0, xMj)*dw; // Delegate-
function argument use
        xM.ColumnsSet(j, xMj);
    }return xM;
}

```

Step 3: declare new VectorD functions **F(...)**, **G(...)** with the return type and signature of the FuncVec delegate:

```

static VectorD F(double t, VectorD X)
{
    return -0.5*(new VectorD(X[0], X[1]));
    // F corresponds to the example
}
static VectorD G(double t, VectorD X)
{
    return new VectorD(X[1], -X[0]);
    // G corresponds to the example
}

```

Step 4: call the method EulerDelegate(...) with the corresponding wrapped functions F and G as arguments:

```

xy = EulerDelegate(x0, new FuncVec(F),
new FuncVec(G), h, n, w);

```

Example. We illustrate the use of these methods on the Itô SDE:

$$dx_t = -0.5x_t dt + y_t dw_t, \quad dy_t = -0.5y_t dt - x_t dw_t.$$

This SDE allows to compare the numerical solution with the analytical solution $x_t = \cos w_t$ and $y_t = \sin w_t$.

The **Euler approximation** of x and y over $\langle 0,1 \rangle$ is computed by the codelines:

```

int n = 100, kernel = 1900317;
double T = 1, h = T/n;
VectorD x0 = new VectorD(0, 1);
MatrixD f, g, xy;
f = MatrixD.CreateFromVector (2,2, -0.5, 0, 0,-0.5);
g = MatrixD.CreateFromVector (2,2, 0, 1,-1, 0);
VectorD wd = Wiener.Create (h, n, kernel);
xy = Euler(x0, f, g, h, n, wd);

```

The following pictures show the generated approximation w of the Wiener process w_t and the approximants x and y of x_t and y_t .

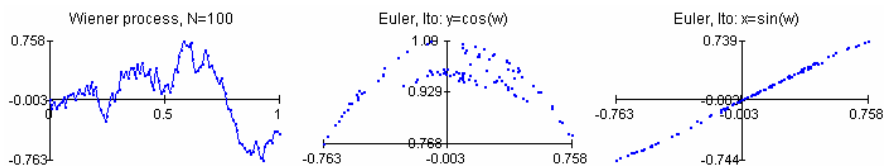


Fig. 5.1. The approximation of $\sin(w_t)$ and $\cos(w_t)$ by the Euler method

The next pictures show the approximants x and y based on the Milstein scheme.

```
MatrixD dg= MatrixD.CreateFromVector (2,2, 0, 1,-1, 0);
xy=Milstein(x0, f, g, dg, h, n, wd);
```

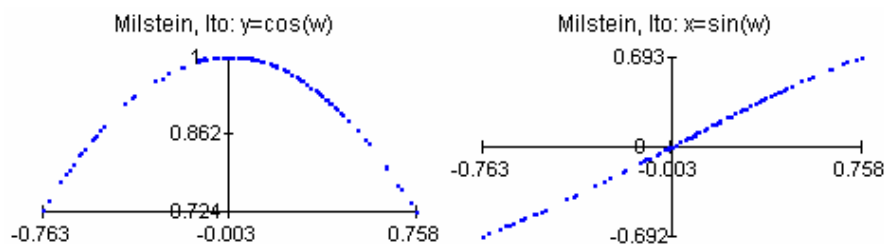


Fig. 5.2. The approximation of $\sin(w_t)$ and $\cos(w_t)$ by the Milstein method

Acknowledgements. This research has been supported by the Czech Ministry of Education in the frame of MSM 0021630529 Research Intention Intelligent Systems in Automation.

References

1. Kloeden P, Platen E, Schulz (1997) Numerical solution of SDE through computer experiments. Springer–Verlag
2. Oksendal B (1995) Stochastic differential equations – An introduction with applications. Springer-Verlag
3. Török Cs (2004) Visualization and data analysis in the MS .NET Framework. Communication of JINR, Dubna, E10-2004-136
4. Török Cs et al. (2002) Professional Windows GUI Programming: Using C#. Wrox Press Inc, Chicago, ISBN 1-861007-66-3
5. Török Cs (1994) Numerical solution of linear stochastic differential equations. Computers & Mathematics with Applications 27(4):1-10

Chapter 6

Visualization and analyses of multi-dimensional data sets using OLAP- A case study of a student administration system

Che-Chern Lin,¹ Fun-Chi Hsu,² Chun-Hung Wu,³ Chia-Jui Hsu,⁴ Chia-Hsun Wu,⁵ Chung-Ping Lee,⁶ Hung-Jen Yang⁷

National Kaohsiung Normal University,
116 Heping First Road, Kaohsiung City, Taiwan (R.O.C.)
ccclin@nknucc.nknu.edu.tw¹, kyohikaru@msn.com²,
hans_wu@hotmail.com³, chiajuimr@hotmail.com⁴, der54367@msn.com⁵,
cl87369@yahoo.com.tw⁶, hjyang@nknucc.nknu.edu.tw⁷

Abstract. In this paper, we introduce the visualization and analyses of multi-dimensional data sets using on-line analytic processing (OLAP). We discuss three fact table schemas for building data cubes for displaying different views and the levels of details of date sets. Five operations of OLAP are used for manipulating data cubes including slice, dice, roll-up, drill-down, and rotation operations. By the five operations one can perform extraction, aggregation, subdividing, changing perspective of information stored in OLAP data cubes. We present a case study to explain the above operations. In the proposed case study, a student administration system was developed. To build data cubes, we established a constellation schema consisting of three fact tables and four dimension tables. We also demonstrate the five operations on the data cubes and discuss how these operations are applied to analyze data.

Keywords. OLAP; Data Cube; Data Warehouse; Multi-dimensional Database

6.1 Introduction

Recently with the development of information technologies, huge data can be handled by computers. Based on data bases, many data analysis techniques are developed and applied in different areas. Data warehouse is one of the popular techniques where multi-dimensional data structures are established by extracting different perspective from related normalized data tables in an operational database. To build a data warehouse, people use data cubes to represent the multi-dimensional data structure. Fact tables are utilized to describe the data structures of the data cubes. With the capabilities of handling and displaying the multi-dimensional data sets, data warehouse applications are successfully utilized in data analyses and decision support systems and therefore widely applied in many fields such as business, engineering, education, medicine, etc. Data warehouses are normally utilized to analyze off-line data for making decisions. On-line analytic processing (OLAP) is a similar technique but focuses on analyzing real time data. It retrieves data from an on-line database. Data mining is another information technique to cluster data by creating association rules. It is also widely applied in different areas such as business, engineering, education, and medicine.

A cubic-wise balance method for privacy preservation has been proposed to furthermore make closely estimation of data without actual data access [1]. MSMiner, an OLAP development software has been generally discussed including design requirements, system architecture, and an application example [2]. Based on unified modeling language (UML), Prat et al. proposed a data warehouse design methodology with three phases: conceptual, logic, and physical phases [3]. A data warehouse application on site determination was implemented by Ahmad et al. where an OLAP-based decision support system was built to select the most appropriate residential area in Miramar, Florida, U.S.A.. Moon et al. discussed the effectiveness on cost estimation of OLAP-based management systems [4]. Efficient view selection methods for data warehouse were proposed to improve the query time and data storage space [5].

In this paper, we apply the data warehouse technique in education and provide a case study on student administration system where OLAP data cubes were generated for manipulating data.

6.2 Design of OLAP cubes

Relational databases use normalized tables where relationships among these tables are used to aggregate data from different data and hence get integrated information for views and reports. A table in a relational database is two-dimensional, consisting of fields (columns) to define data attributes of the table and records (rows) to describe the field values for a particular data instance. Normalizations are utilized to generate the normalized tables in a relational database from an original unnormalized data set. During normalization procedure, tables are sequentially divided into smaller tables without losing information. After normalizations, tables are said to be in normal forms. Five normalizations are used to generate five normal forms from first normal form (1NF) to fifth normal form (5NF). To implement the relationships among these tables in a database, people use join operations to integrate different data from different tables using “keys”. Structured query language (SQL) is used to practically realize the join operations using a “select” command. Two keys are generally used in tables: primary key (PK) and foreign key (FK). A PK is used to identify a particular record and to get the rest of field values associated with this particular record. A FK is utilized to link the PK of another table to get more information from the linked table.

A table in a relational database only shows the information related to the table. Using SQL commands gets some partially integrated data. For decision making, people need total information to analyze data and make decisions. Data warehouses are applied to integrate data from a database in support of subject-oriented data analyses for decision making. Figure 6.1 shows a data warehouse process model [6]. In the figure, a data warehouse obtains its data by using an ETL (extract, transform, and load) routine from different ways including external data, operational database and independent marts. To build a data warehouse, people use fact tables to aggregate different data from dimension tables which provide necessary information for establishing the data warehouse. Similar to a PK, a dimension key is used to link a fact table to a dimension table associated with the dimension key. Three types of fact table schemas are often used to build data cubes including: star, snowflake, and constellation schemas. The star schema is the simplest structure of fact table where a single fact table is located at the center of the structure, linking each dimension table with a corresponding dimension key. To decrease data redundancy, a snowflake schema is used to further divide some dimension tables. A constellation schema is another structure where more than one fact table is located at the center of the

structure. In a constellation schema, dimension tables can be multiply linked to fact tables.

Based on the techniques of building data warehouses, OLAP is a query-based approach to handle real time data sets for data analyses. Five operations are often applied to perform OLAP including: slice, dice, drill-down, roll-up, and rotation operations. A slice is an operation to perform data retrieval on a single dimension of an OLAP cube. A dice operation is similar to slice operation but focuses on selecting data on two or more dimensions. A roll-up operation performs data generalization by aggregating data on one particular dimension on an OLAP cube. A drill-down operation, a reverse process of roll-up, is used to obtain more detailed data for a particular dimension on an OLAP cube. A rotation simply selects another perspective of an OLAP cube by rotating the displayed dimension without aggregation, sub-dividing, and extraction of the data.

6.3 Case study

To demonstrate visualization of multi-dimensional data sets, we present an example of student administration system. Figure 6.2 shows the schema of the proposed example where a constellation schema is used to construct OLAP cubes. Three fact tables are used to form the center parts of the constellation schema including course selection table, score table, and absenteeism table. In addition to the fact tables, four dimension tables are also utilized to form the OLAP cubes including teacher, class, course and student tables. The detailed fields of the fact tables and dimension tables are also displayed in Figure 6.2.

6.4 Conclusion

We introduced the visualization and analyses of multi-dimensional data sets using OLAP. We also discussed three fact table schemas for building data cubes. Five operations are discussed for manipulating data cubes including slice, dice, roll-up, drill-down, and rotation operations. We presented an example of a student administration system to explain the above operations. To demonstrate the example, we established OLAP cubes using a constellation schema.

As about further studies, we suggest extending the system with more tables and building more complex schema to get more information.

References

1. Liu Y, Sung SY, Xiong H (2000) A cube-wise balance approach for privacy preservation in data cubes. *Information Sciences* 176:1215-1240
2. Shi Z, Huang Y, He Q, Xu L, Liu S, Qin L, Jia Z, Li J, Huang H, Zhao L (2007) MSMiner- a developing platform for OLAP. *Decision Support Systems* 42:2016-2028
3. Prat N, Akoka J, Comyn-Wattiau I (2006) A UML-based data warehouse design method. *Decision Support Systems* 42:1449-1473
4. Ahmad I, Azhar S, Lukauskis P (2004) Development of a decision support system using data warehousing to assist builders/developers in site selection. *Automation in Construction* Vol. 13:525-542
5. Moon SW, Kim JS, Kwon KN (2007) Effectiveness of OLAP-based cost data management in construction cost estimate. *Automation in Construction* 16:336-344
6. Roiger RJ, Geatz MW (2003) *Data mining: A tutorial-based primer*. Addison Welsley, Pearson Education, Inc., Chapter 6

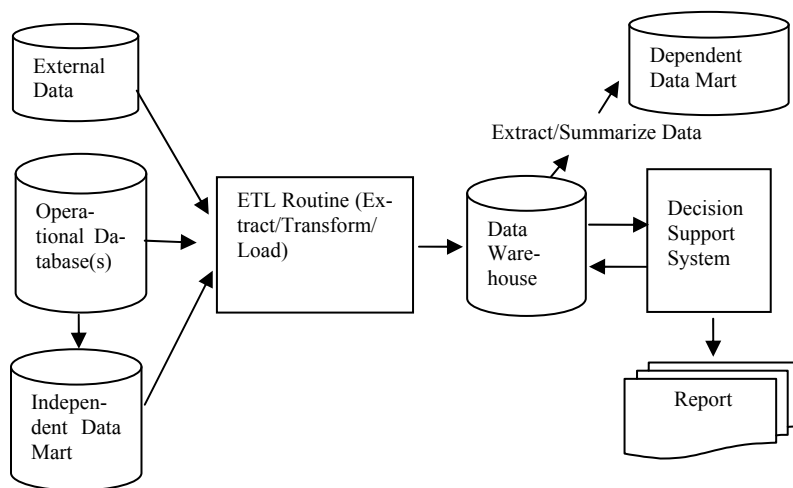


Fig. 6.1. A data warehouse process model (taken from [6])

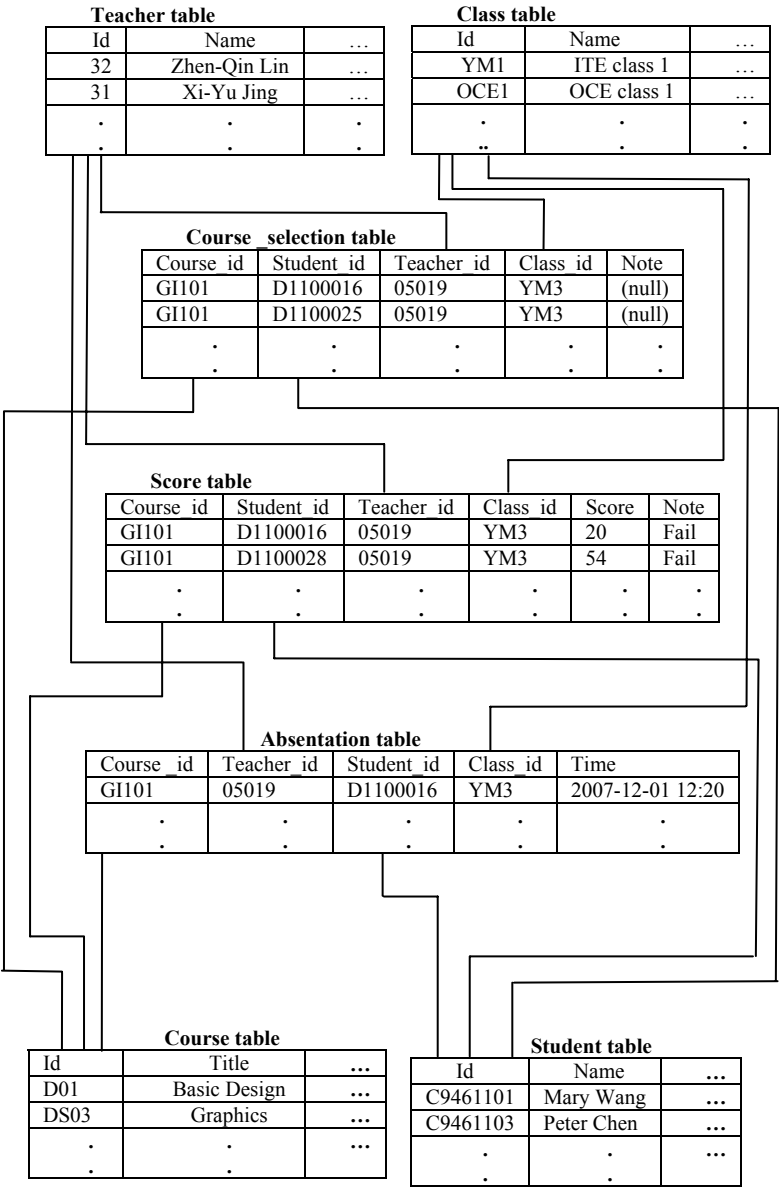


Fig. 6.2. The constellation schema of the proposed OLAP cubes

Chapter 7

Design of a computer-based triangle chess machine

Che-Chern Lin,¹ Chia-Jung Hsu,² Ming-Jun Chan,³ Chih-Hung Chung,⁴
Chong-Cheng Hong,⁵ Hung-Jen Yang⁶

National Kaohsiung Normal University,
116 Heping First Road, Kaohsiung, TAIWAN (R.O.C)
ccclin@nknucc.nknu.edu.tw¹, z3charles@msn.com²,
taiwanka@yahoo.com.tw³, bbc160@yahoo.com.tw⁴,
t771580@yahoo.com.tw⁵, hjyang@nknucc.nknu.edu.tw⁶

Abstract. Computer-based chess machines have been built based on tree searches and parallel computing. Several papers have discussed the computer implementations of different types of chess games such as chess (western), Chinese chess (Xiangqi), Japanese chess (Shogi), and Hex. This paper focuses on the implementation of triangle chess machine. We first make a brief explanation on the popular chess games and their related studies. We then describe the rules for playing a triangle chess. Basically, the triangle chess machine proposed in this paper considers all possible configurations of chess patterns and then makes moves based on these patterns. We present the software modules of the machine and discuss the computation issues related to this study. Conclusions and the directions for future studies are also given at the end of this paper.

Keywords. Triangle Chess, Chess Machine, Pattern Marching, Searching Space, Computer Chess

7.1 Introduction

Recently with the promotion of computational speed and data storage, computers have stronger capability to handle more complicated computing. Computer programs for different types of chess games have been built successfully in the past decades including the developments of chess (western), Chinese chess (Xiangqi), Japanese chess (Shogi), Hex, and Go. Xiangqi, a Chinese chess, is very popular in worldwide Chinese community. It uses an 8x8 grid-like chessboard played by two players, “*Black*” and “*Red*” [1]. Each of them owns a territory in a roughly half size of the chessboard and has fifteen pieces. These pieces are categorized into seven different types of roles with differ moving regulations, respectively. The seven roles include King, Assistant, Elephant, Horse, Chariot, Gunner, and Pawn [1]. The player who eliminates all of the opponent’s pieces wins the game. Similar to Xiangqi, Shogi is a Japanese chess played by two players, called “*Black*” and “*White*” [2]. It uses a 9x9 grid-like chessboard. Each of players has 20 pieces. Hex is also a two-player game played on a hexagonally grid-like chessboard with regularly an 11x11 rhombus [3]. The two players, named “*Red*” and “*Blue*”, have their own sides, respectively. During a Hex game, each player uses his pieces to make a link from his side to the opponent’s side. The first player to make a successful implementation of the link wins the game. Go is another popular chess in East Asia. It also has two players, “*Black*” and “*White*”, played on an 18x18 grid-like chessboard [4]. During a Go game, each player uses his pieces to form his own territory by removing the opponent’s pieces tightly surrounded by his pieces [4]. After an endgame, the player who has the maximum area of territory wins the game.

Basically a chess machine uses searching and parallel computing techniques. However, it requires computers with faster CPU speed and more memory capacity. Using rule-based machines needs to transfer the chess experts’ strategies into rules by which to determine the next steps. It is sometimes difficult to generate and implement these rule-based chess machines.

Originally derived from Deep Thought, Deep Blue, developed at IBM Watson Research Center, is a successful computer-based chess machine which defeated a World Chess Champion in 1997 [5]. It basically used parallel computing techniques and searching trees to deal with massive data processing. The computational algorithms developed for Deep Blue calculate all possible moves and might result in massive data manipulations. After calculating the possible moves, Deep Blue then uses evaluation functions to measure the possible scores for these moves and makes a decision

for next steps. Deep Blue not only used the computer software techniques but also hardware techniques. It is so far the most successful chess machine implemented by computer technologies. Iida et al. compared the similarity and the differences between Chess and Shogi, and described the techniques used to build a computer-based shogi machine, including opening and endgame plays, searching algorithm, tactical strategies, and position evaluation. [2]. Wu and Beal used endgame database construction method to implement a computer-based Chinese chess machine and demonstrated its performance in reducing the size of the database [1]. Anshelevich presented a hierarchical method for building a Hex chess machine where he used deduction rules to recursively compute the Hex positions from the beginning of a game [6].

In this paper, we present a case study of implementation of a computer-based triangle chess machine. We also discuss the rules for playing a triangle chess game and the design methodology of this chess machine. Finally we give conclusions and suggest the directions for future studies.

7.2 Implementation

The rules for playing a triangle chess are very simple and straightforward. Figure 7.1 shows a triangle chess machine where fifteen circles are arranged in a triangle chessboard. During a triangle chess game, one should select one or more circles (up to five circles) in each step. The selected circles in each step must form a line. This is called “*co-linear*” rule. The player who selects the last circle is the loser.

We used Visual C++ to implement the triangle chess machine. The machine provides a user interface for users to play a triangle chess game and a replay function to repeat the whole process of the game. To implement the machine, we need some software modules (subroutines) to perform the functionalities. There are mainly four modules for doing this. The first module is to validate if the circles selected by a user (the opponent) fit the co-linear rule. We then call this module “co-linear checking” module. The second module is to determine the pattern of the configuration of the triangle chess after the opponent selects his circles. The second module also makes a responding selection of circles (automatically done by the machine) to react the opponent selection. This module is called pattern determination/circles selection module. The third module is called replay module serving to record all steps in a triangle game and to perform replay functionality if necessary. The fourth module is called “user interface” module providing a graphic interfaces for playing a triangle chess game.

Figure 7.2 shows the four functional modules and their relationships. The four modules are explained in details below.

1. Co-linear checking module: The co-linear checking module uses tree structures to determine if circles selected by the opponent are co-linear. Each circle in a triangle chessboard has its own searching tree by which to examine if selected circles satisfy the co-linear rule. Each searching tree describes all possible co-linear paths for its corresponding circle. If the opponent selects his circles, the module examines if the selection of circles is on any of co-linear paths. Figure 7.3 shows a triangle chess configuration and the associated tree structures. For convenience, we number the 15 circles in a triangle chessboard from 1 to 15, as indicated in Figure 7.3(a). By the property of symmetry, the 15 circles are categorized into 4 types based on the geometrical property. They are shown as follows:

Type 1: circles 1, 11, and 15.

Type 2: circles 2, 3, 7, 10, 12, and 14.

Type 3: circles 4, 6 and 13

Type 4: circles 5, 8 and 9.

The sampled examples for the four types of trees are also shown in Figure 7.3. It is important to note that the selected circles must be sorted before using a tree since the user (opponent player) may make their selections of circles in any orders.

2. Pattern determination/circles selection module: This module stores all possible configurations of patters for a triangle chess game. There are 15 circles in a triangle chessboard and each of circles has two stages: selected (= 1) and unselected (= 0). We then have 2^{15} (=32768) combinations of chess configurations. Of the 32768 combinations, only 7795 patterns satisfy the co-linear rule. We therefore use the 7795 patterns as pattern-matching data source. Each of patterns is associated with a response configuration which determines the next move (done by the machine). Since the values of the 15 circles are binary (0 and 1), we then encode each of patterns in a two-byte short integer. The response patterns are also encoded in two-byte short integers. We use a two-dimensional array to store these short integers
3. Replay module: The replay module records all steps played in a game and replays them if a user (opponent) needs to know the history of the game.
4. User Interface module: The user interface module provides a playing platform for an opponent player.

7.3 Conclusions

We made a brief explanation on several popular chess games and their computer implementations. We also presented a case study of implementation of a computer-based triangle chess machine. Basically, the chess machine considers all possible configurations of chess patterns and then makes a selection of circles based on these patterns. We demonstrated the software modules of the machine and discussed the issues related to this study.

Using decision rules to implement chess machine might be an interesting issue for future studies since it can significantly reduce searching spaces.

References

1. Wu R, Beal DF (2001) Solving Chinese chess endgames by database construction. Information Sciences 135:207-228
2. Iida H, Sakuta M, Rollason J (2002) Computer Shogi. Artificial Intelligence 134:121-144
3. [http://en.wikipedia.org/wiki/Hex_\(board_game\)](http://en.wikipedia.org/wiki/Hex_(board_game))
4. [http://en.wikipedia.org/wiki/Go_\(board_game\)](http://en.wikipedia.org/wiki/Go_(board_game))
5. Campbell M, Hoane Jr AJ, Hsu F (2002) Deep Blue. Artificial Intelligence 134:57-83
6. Anshelevich VV (2002) A hierarchical approach to computer hex. Artificial Intelligence 134:101-120
7. Müller M (2001) Global and local tree search. Information Sciences 135:187-206
8. Ewerhart C (2002) Backward induction and the game-theoretic analysis of chess. Game and Economic Behavior 39:206-214

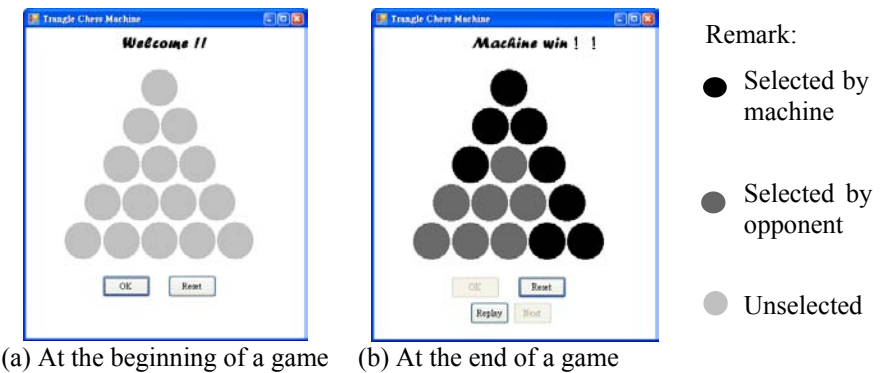


Fig. 7.1. The triangle machine

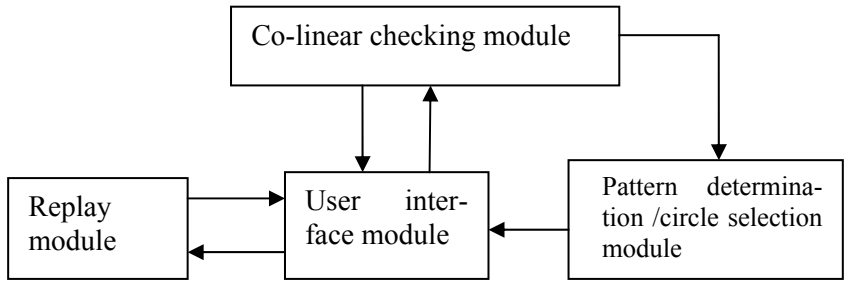


Fig. 7.2. The functional modules in the chess machine

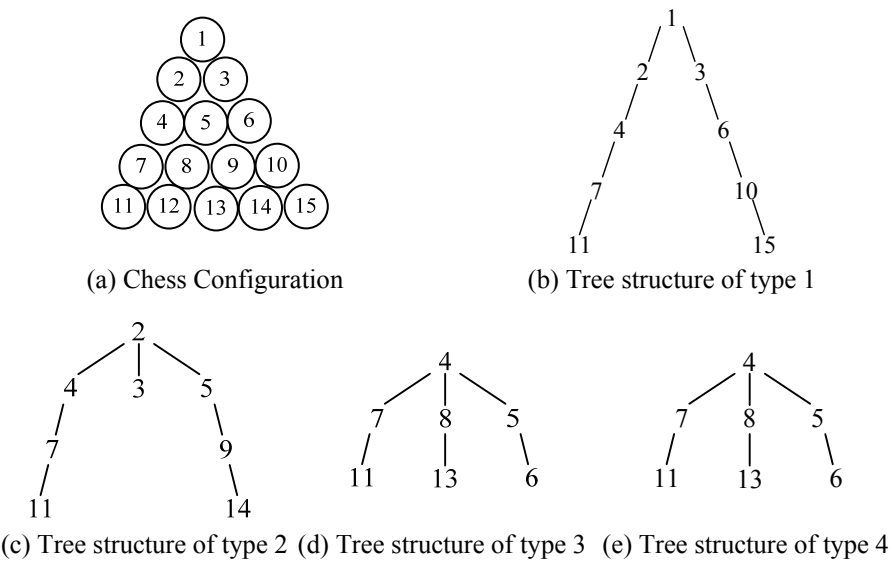


Fig. 7.3. Trees to examine the co-linear rule

Chapter 8

A taxonomy model for analyzing project-based learning using Analytical Hierarchy Process (AHP): A case study of design of a beer company website in an elementary school

Chung-Ping Lee,¹ Che-Chern Lin,² Rong-Jyue Fang,³ Howard Lo,⁴ Ming-Chih Wu⁵

^{1,2,4} Department of Industrial Technology Education,
National Kaohsiung Normal University, Taiwan (R.O.C.)
cl87369@yahoo.com.tw,¹ cclin@nknucc.nknu.edu.tw,²
wardlab@yahoo.com⁴

ho-

³ Department of Information Management,
Southern Taiwan University of Technology, Taiwan (R.O.C.)
rxf26@mail.stut.edu.tw

⁵ Graduate Institute of Educational Technology,
National Pingtung University of Education, Taiwan (R.O.C.) michy-
wu@yahoo.com.tw

Abstract. This paper presents a taxonomy model which applies analytical hierarchy process (AHP) to project-based learning. To implement the proposed model, a website has been built for the sixth-graders of an elementary school for their project-based learning. The learning target is the knowledge about beer. The learning materials were provided by a local beer company. Six factors are considered as the inputs of the AHP including company introduction, characteristics of beer of the company, company visiting, beer promotion, beer manufacturing, and learning activity sharing. To investigate the importances among the six factors, we interviewed five experts who have at least seven years of teaching experience in an elementary school in Taiwan. The AHP is utilized to generate the weights (importances) for the input variables for the taxonomy model.

Keywords. Project- based learning, PBL, Analytical Hierarchy Process, AHP

8.1 Introduction

Analytical hierarchy process (AHP) is developed for decision-making [1-4]. Basically, an AHP model is hierarchically layered structure. Figure 8.1 shows a three-layer AHP [3]. Nodes are used in each layer to represent factors or dimensions. One utilizes weights to link two nodes in the adjacent layers. Mappings are performed from inputs to outputs sequentially. The mapping function is given by the follow formula [3]

$$y_j = \sum_{i=1}^n w_{ij}x_i \tag{8.1}$$

where

- y_j = the output for node j in a particular layer,
- x_i = the output of node i in the previous layer of this particular layer,
- w_{ij} = the weight linking node j in the particular layer and the node i in the previous layer,
- n = the number of node in the previous layer.

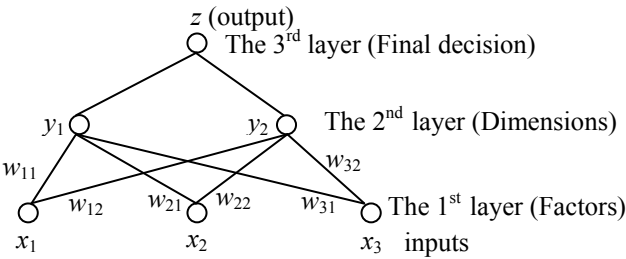


Fig. 8.1. A simple three-layer AHP (taken from [3])

Project-based learning (PBL) is a model that organizes learning materials from different projects. The projects are complex tasks that involve learners in design, problem-solving, decision making, investigative activities, etc [5]. Instructors provide learners some activities to analyze information on learning topics, compare other cases with their problems, and develop an optimal method for the task [6]. The task-oriented approach is used to find directions of problem solving by providing different practical

problems. Finally, learners must gather and organize relevant information in order to implement their learning projects.

In this paper, we propose a taxonomy model which applies a two-layer AHP to PBL. We introduce the AHP procedure and explain how AHP is utilized to PBL on beer. A website whose contents were provided by a local beer company has been built for the sixth-graders of an elementary school for their learning on beer. Six factors were determined using a brainstorming procedure including company introduction, characteristics of beer of the company, company visiting, beer promotion, beer manufacturing, and learning activity sharing. Five experts were selected to answer specially designed questionnaires. Finally we present the evaluation model for the taxonomy.

8.2 The AHP procedure

The AHP procedure of the propose model is depicted as follows [1-4, 7].

Step 1: Define the problem and determine the goal

The goal of this study is to select the useful messages and knowledge from the forum of the website for PBL on beer. This is done by collection of information through internet, group discussion, and special homework related to this study.

Step 2: Select factors for the model

We used brainstorming to determine the relative importance for establishing evaluation criteria of the elements in the hierarchy. Nine sixth-graders in an elementary school were selected to carry out the brainstorming and generate the structure of factors using a free software package, Xebece. Xebece produces structures of brainstorming by the combination of catalogues, attachment of documents, and XML documents [8]. After the brainstorming, six factors were considered to build the proposed model including company introduction, characteristics of beer of the company, company visiting, beer promotion, beer manufacturing, and learning activity sharing.

Step 3: Design the questionnaire

The questionnaire was designed to make all possible pair-wise comparisons among the factors. Table 8.1 shows a typical nine-point scale for an AHP questionnaire [3, 4]. The questionnaire was designed to measure all possible importance ratios among the factors. Table 8.2 shows a simple example of the questionnaire where three factors are selected: factors A, B, and C. In the table, factor A is twice important than factor B, the ratio

of factors A to B is 2:1. Row 1 is corresponding to the ratio of factors A to B. In row 1, we then mark “√” in the cell associated with a value of 2 (closed to the factor of A). Similarly, the importance ratio of factor A to factor C is 3: 1. The importance ratio of factor B to factor C is 1:5. Both of them are shown in Table 8.2 (rows 2 and 3).

Table 8.1. The definition and explanation of AHP 9-point scale (taken from [3, 4])

Intensity of Relative Importance	Definition
1	Equal Importance
3	Moderate important of one over another
5	Essential or strong important
7	Demonstrated importance
9	Absolute importance
2, 4, 6, 8	Intermediate values between the two neighboring scales

Table 8.2. A simple example of questionnaire (taken from [3])

Factor	9	8	7	6	5	4	3	2	1	2	3	4	5	6	7	8	9	Factor
A (row 1)								√										B
A (row 2)							√											C
B (row 3)													√					C

Step 4: Use the questionnaire to collect the experts’ opinions.

After collecting questionnaires from experts, one uses a “*matrix of importance ratios*” to describe the results of pair-wise comparisons. Eq. 8.2 shows the matrix of importance ratios associated with Table 8.2 [3]. The matrix is a symmetrical and reciprocal matrix due to pair-wise comparisons.

$$\begin{matrix} & \begin{matrix} A & B & C \end{matrix} \\ \begin{matrix} A \\ B \\ C \end{matrix} & \begin{bmatrix} 1 & 2 & 3 \\ 1/2 & 1 & 1/5 \\ 1/3 & 5 & 1 \end{bmatrix} \end{matrix} \tag{8.2}$$

Step 5: Test the consistency

The Consistency Index (CI) is utilized to express the degree of consistency. Saaty [4] defined the consistency index (CI) as follows:

$$CI = \frac{\lambda_{\max} - n}{n - 1} \tag{8.3}$$

where $\lambda_{\max}$ is the maximum eigenvalue of the matrix of the importance ratios and n is the number of factors.

Accordingly, Saaty [4] defined the Constituency Ratio (CR) as follows:

$$CR = \frac{CI}{RI} \quad (8.4)$$

where Random Index (RI) is given by Table 8.3. If the value of the consistency ratio CR is less than or equal to 0.1 the questionnaire is considered to be acceptable. If CR is greater than 0.1, the questionnaire fails.

Table 8.3. Random Index [1]

n	1	2	3	4	5	6	7	8	9	10
RI	0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

Remark: n is the number of fact

8.3 Experimental setup and results

In this study, five experts from an elementary school in Taiwan were selected. All of them have at least seven years of teaching experience. The results of five experts' questionnaires and consistency tests are shown in Table 8.4. Only two questionnaires (Expert 2 and Expert 4) passed the consistency tests.

Table 8.4. Eigenvalue and consistency indices of experiment

Experts	$\lambda_{\max}$	CI	CR
Experts 1	7.15	0.23	0.19
Experts 2	6.00	0	0.00*
Experts 3	7.30	0.26	0.21
Experts 4	6.33	0.07	0.05*
Experts 5	7.18	0.24	0.19

Remark: * indicates $CR \leq 0.1$ (passing the consistence tests)

We took geometrical averages of the weights from the questionnaires which passed the consistency tests.

The evaluation formula for the taxonomy model is obtained by taking the normalized eigenvector associated with the maximum eigenvalue of the matrix of importances ($\lambda_{\max}$). Eq. 8.5 demonstrates the evaluation formula

$$y = 0.071x_1 + 0.106x_2 + 0.232x_3 + 0.068x_4 + 0.228x_5 + 0.228x_6 \quad (8.5)$$

where

- x_1 = company introduction,
- x_2 = characteristics of beer of the company,
- x_3 = company visiting,
- x_4 = beer promotion,
- x_5 = beer manufacturing,
- x_6 = learning activity sharing.

In Eq. (8.5), the higher value of y means the stronger recommendation for taxonomy.

8.4 Conclusions

We proposed a taxonomy model which applied AHP to PBL. The learning target of the PBL is the knowledge of beer. The learners were the sixth-graders of an elementary school. To implement the model, we built a web-site cooperating with a local beer company. Six factors were determined using a brainstorming procedure including company introduction, characteristics of beer of the company, company visiting, beer promotion, beer manufacturing, and learning activity sharing. Five experts were selected to answer the specially designed questionnaires. Finally we used the AHP procedure to generate the evaluation formula for the taxonomy model.

As about the directions of further studies, Delphi technique may be considered to collect more experts' opinions into the project. Besides, using fuzzy logic to describe the factors might be an interesting topic for future study. Finally, we suggest using more layers in the AHP structure to collect more detailed items for further analyses.

References

1. Saaty TL (1990) How to mark a decision: the analytic hierarchy process. *European Journal of Operational Research* 48(1):9–26
2. Saaty TL (2003) Decision-making with the AHP: Why is the principal eigenvector necessary. *European Journal of Operational Research* 145:85–91
3. Huang C, Lin Y, Lin C (2007) An evaluation model for determining insurance policy using AHP and fuzzy logic: case studies of life and annuity insurances. In: *The 8th WSEAS international conference on fuzzy systems*, Vancouver, Canada, pp 126–131
4. Saaty TL (1980) *The Analytic Hierarchy Process*. Revised and extended 1988, McGraw-Hill, New York

5. John TW (2000) A review of research on project-based learning. T. A. Foundation, California
6. Brunetti AJ, Petrell RJ, Sawada B (2003) Team project-based learning enhances awareness of sustainability at the University of British Columbia, Canada. *International Journal of Sustainability in Higher Education* 4(1):210-217
7. Lee SK, Yoon YJ, Kim JW (2007) A study on making a long-term improvement in the national energy efficiency and GHG control plans by the AHP approach. *Energy Policy* 35:2862–2868
8. Kennke R (2007) Xebece: Visualize and organize information easily. Retrieved 6.1, from <http://xebece.sourceforge.net/>

Chapter 9

A framework of mixed model of a neural network and boolean logic for financial crisis prediction

Che-Chern Lin,¹ Ghuan-Hsien Mao,² Howard Lo,³ Hung-Jen Yang⁴

^{1,3,4} National Kaohsiung Normal University,
116 Heping First Road, Kaohsiung City, Taiwan (R.O.C.)
² Shu-Te University,
59 Hun Shan Road, Yen-Chau, Kaohsiung County, Taiwan (R.O.C.)
cclin@nknucc.nknu.edu.tw,¹ js.mao@msa.hinet.net,²
howardlab@yahoo.com,³ hjyang@nknucc.nknu.edu.tw⁴

Abstract. This paper proposes a framework for pattern classifications. This framework includes two neural networks and a logic circuit. The first neural network serves to classify regular instances, and the second one handles the overlapped instances caused by the first neural network. The proposed framework integrates the two neural networks by the logic circuit and then makes the final classifications. We provide two methodologies using the proposed framework including financial crisis prediction and principle components analysis. In the financial crisis prediction case, we use the financial ratios of individual companies as the inputs of the first neural network and the macroeconomic indicators as the inputs of the second neural network. In the principle components analysis case, we use the attributes in a major factor group as the inputs of the first neural network and those in a minor factor group as the inputs of the second neural network.

Keywords. Neural Networks, Logic Circuit; Financial Crisis Prediction

9.1 Introduction

Traditionally, people use statistical methods to analyze data, build models, and make decisions (prediction, estimation, and classification). Instead of statistical methods, recently artificial intelligent algorithms are widely utilized in different areas. The most popular artificial intelligence algorithms include data mining algorithms, fuzzy systems, genetic algorithms, and neural networks. Data mining algorithms use computer programs to find relationships among different attributions by establishing association rules. Based on an iterative procedure to find the optimal combinations of items, the apriori algorithm is the most popular one to general association rules for categorical items. Basket analyses are successful applications for business to analyze customers' behaviors. Fuzzy expert systems mimic the thinking processes of human beings where verbal expresses are used to generate rules and make decisions. Unlike traditional crisp values, fuzzy variables use membership functions to describe the marching degree of how inputs fit a particular fuzzy set. Two types of membership functions are often adopted in fuzzy systems: triangular and trapezoidal functions. In general, a fuzzy expert system consists of three parts: a fuzzy base, an inference engine, and user interfaces. The fuzzy base consists of a set of fuzzy rules usually obtained by domain experts. The inference engine is the mechanism to make conclusions by firing the fuzzy rules. The user interfaces allow users to define membership functions, compose fuzzy rules and display outcomes. Genetic algorithms use an evolutionary procedure to find optimal solutions. Like biological evolutionary process, genetic algorithms use fitness function to evaluate the performance for particular generations and use genes (chromosomes) as input strings. In genetic algorithms, selections are used to get better offspring and mutations are utilized to mimic the random changing of genes. Neural networks use computer programs to simulate the structure of human brains and the cognition processes of human beings. A neural network is usually a layered structure with nodes in each layer. Weights are used to connect two nodes between two adjacent layers and updated by training algorithms. The back-propagation algorithm is one of the most popular updating algorithms where the gradient descent method is utilized to minimize the errors. The training algorithm uses a set of well-known examples, called training set, to backwardly adjust weighs from outputs to inputs. This procedure is called training procedure. After the training procedure, neural networks are ready to classify unknown instances based on the well-trained weights. Neural networks are widely applied to financial applications. A lots of studies related to financial crisis prediction can be found in [1-6].

9.2 The Framework

Figure 9.1 shows the diagram for the proposed framework. There are two neural networks and a logic circuit in Figure 9.1. The first neural network functions serve to classify regular instances and the second one handles the overlapped instances caused by the first neural network. For a two-class classification problem (Class I and Class II), the classification results are determined in Table 9.1.

Table 9.1. The classification results of a two-class classification problem

A	B	Class assigned
0	0	Undetermined
0	1	Class I
1	0	Class II
1	1	Overlapped

To handle the overlapped instances, we use the second neural network to furthermore classify these overlapped instances. We select these overlapped instances as the training data for the second neural network. The logic circuit is used to integrate the two neural networks and get the final classification results. Table 9.2 is the truth table for the logic circuit of the proposed framework.

Table 9.2. The truth table for the logic circuit

Input			Output
A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Remark: Y=1 represents Class I, and Y= 0 represents Class II.

The simplified Boolean expression for Table 9.2 is given by

$$Y = B(\bar{A} + C) \quad (1)$$

The logic circuit for Eq. 9.1 is also shown in Figure 9.1.

In brief, the procedure of the proposed framework is described as follows:

- Step 1: Train the first neural network. The overlapping instances are the patterns with $A = 1$ and $B = 1$.
- Step 2: Train the second neural network using the overlapping instances obtained in Step 1.
- Step 3: Implement the classifier using the proposed framework shown in Figure 9.1.

9.3 Methodologies

In the proposed framework, the first neural network serves to perform the main classifications and the second one handles the overlapped patterns to improve the classification accuracies. We then call the first neural network “*primary neural network*” and the second one “*secondary neural network*”. Below, we present two methodologies applying the proposed framework.

Methodology 1: Establish the primary and secondary neural networks using well-defined dimensions of attributes. If the dimensionality of the variables has well-defined, one can use the variables in major dimension as the inputs of the primary neural network, and the minor dimension as the input of the secondary neural network. Financial crisis prediction is one of the applications of using methodology 1 [6]. Previous studies have indicated the financial crisis can be predicted using neural networks based on two groups of variables: financial ratios for individual companies and macroeconomic indicators. The financial ratios affect the financial crisis more strongly than the macroeconomic indicators. We then use the financial ratios as the inputs for the primary neural network and the macroeconomic indicators as the inputs for the secondary neural network. Mao used this framework to predict the financial crises for public electronic companies in Taiwan and got better results than regular neural networks [6]. The financial ratios and macroeconomic indicators used in Mao’s study are shown in Table 9.3 [6].

Methodology 2: The second methodology is to use statistical techniques to separate variables into two groups according to their importance: the major affecting and minor affecting groups before using the proposed framework. Consider a set of instances with n attributes. Based on the importance of the affecting degree of the n attributes, one can use statistical methods to divide the n attributes into two groups: the major factor group with higher affecting degrees and minor factor group with lower affecting degrees. Principle components analysis is one of the popular methods to do this. After dividing the input attributes, one can use the attributes in major

factor group as the input of the primary neural network and those in minor factors as inputs of the secondary neural network.

9.4 Conclusions

We introduced the artificial intelligence algorithms and their applications. We proposed a framework for pattern classifications. This framework includes two neural networks and a logic circuit. The first neural network serves to classify regular instances, and the second one handles the overlapped instances caused by the first neural network. The proposed framework integrates the two neural networks by the logic circuit and then makes the final classifications. We provided two methodologies using the proposed framework including financial crisis prediction and principle components analysis. In the financial crisis prediction case, we used the financial ratios of individual companies as the inputs for the primary neural network and the macroeconomic indicators as the inputs for the secondary neural network. In the principle components analysis case, we used the attributes in major factor group as the input of the primary neural network and those in minor factors as the inputs of secondary neural network.

As for the further studies, people might add fuzzy logics into the proposed framework to furthermore classify overlapped patterns. Using more neural networks for the framework might be interesting research topics for further works.

References

1. Altman EI (1968) Financial ratios, discriminant analysis and the prediction of Corporate bankruptcy. *Journal of Finance*, pp 589-609
2. Altman EI, Marco GV, Varetto F (1994) Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks (The Italian Experience). *Journal of Banking and Finance* 18:505-529
3. Beaver WH (1966) Financial ratios as predictors of failure. *Journal of Accounting Research*, pp 71-111
4. Blum M (1974) Failing company discriminant analysis. *Journal of Accounting Research* 12:1-25
5. Coats PK, Fant LF (1993) Recognizing financial distress patterns using a neural network tool. *Financial Management* 12:142-155
6. Mao C (2002) A study on financial crisis forecasting using a mixed model of neural network and digital logic manipulation — A case study of public elec-

tronic companies. Master Thesis, Shu-Te University, Kaohsiung County, Taiwan

Table 9.3. The financial ratios and macroeconomic indicators used for financial crisis prediction [6]

Financial ratio	Macroeconomic indicators
Return on total assets	Gross Domestic Product
Growth rate in sales	Export
Growth rate in operating profits	Import
Growth rate in total assets	Annual increasing rate of wages
Growth rate in equity	Employment rate
Growth rate in fixed assets	Unemployment rate
Current ratio	Economic growth rate
Quick ratio	Annual increasing rate of loans and investments
Debt ratio	Rediscount rate
Times interest earned ratio	Interest rate in money market
Accounts receivable turnover	Exchange rate
Inventory turnover	
Total asset turnover	
Debt- to-Equity Ratio	

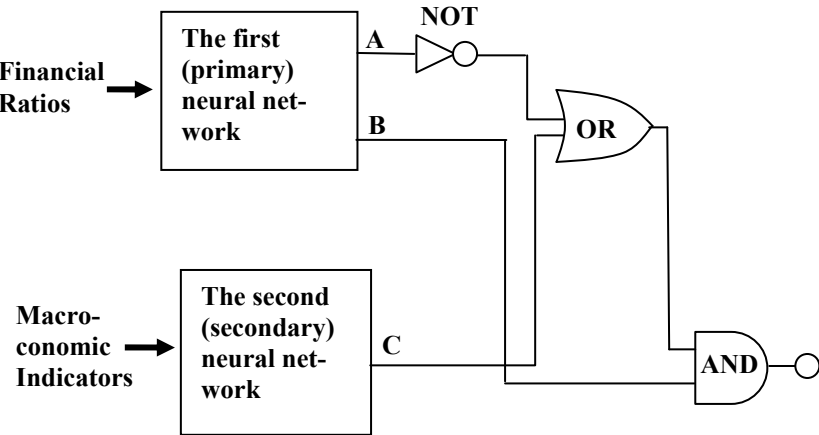


Fig. 9.1. The proposed framework

Chapter 10

Chinese document clustering using self-organizing map-based on botanical document warehouse

Howard Lo,¹ Che-Chern Lin,² Rong-Jyue Fang,³ Chungping Lee,⁴ Yu-Chen Weng⁵

^{1,2,4,5} Department of Industrial Technology Education, National Kaohsiung Normal University, TAIWAN

³ Department of Information Management, Southern Taiwan University of Technology, TAIWAN

13F., No.225, Guanghua 3rd Rd., Kaohsiung City 806 Taiwan

howardlab@gmail.com,¹ cclin@nknucc.nknu.edu.tw,²

rxlf26@mail.stut.edu.tw,³ cl87369@yahoo.com.tw,⁴ jane2080@gmail.com⁵

Abstract. The exponential growth of information has made an overflow situation in the sea of information. It had created difficulties in the search for information. An efficient method to organize the query of information and assist users' navigation is therefore particularly important. In this paper, we applied SOM algorithm to cluster Chinese botanical documents onto a two-dimensional map. Each botanical document has been regarded as bags of words, and transferred into a plain text respectively. We applied term frequency and inverse term frequency to extract key terms from documents as the input of SOM. 892 Chinese botanical documents have been projected onto a 2D map to assist users' navigation. In our experimental results, the lowest recall was 0.71 for Polygonaceae documents and the highest recall rate was 0.94 for Amaranthaceae documents. The lowest precision rate was 0.81 for Umbelliferae documents, and the highest precision rate was one hundred percent for Convolvulaceae and Cruciferae documents.

Keywords. Self Organizing Map, Information retrieval, Vector Space Model, Clustering, Browsing

10.1 Introduction

The Self-Organizing Map (SOM) algorithm, which is developed by Kononen[1], is largely use in images and documents clustering. The SOM algorithm works best with browsing tasks, especially in a very large document storage, such as an document warehouse [2].

The aim of this study is to apply SOM algorithm to cluster those botanical data in a document warehouse into a two-dimensional map, in order to fulfill the naïve users' information needs. We exploited the discrimination values to cull important terms from the document warehouse, and transferred the unstructured documents into vectorized tuples as inputs of the SOM algorithm. The results will be projected onto a two-dimensional SOM map.

When considering retrieval performance evaluation, we adopted the recall rate and precision rate as our measurements.

The rest of the paper is organized as follows: section 10.2 describes related works and the SOM algorithm. Section 10.3 describes the research steps and model of our research. The evaluation and experiment results would be discuss in section 10.4, and section 10.5 will come to a conclusion and point out the future research.

10.2 Related works

According to the survey of Sullivan[3], eighty percent of the necessary information for the decision maker were came from unstructured data, such as documents and only twenty of them were came from structured data, such as a database. There is a need to deal with documents.

10.2.1 Dealing with unstructured documents

Since 1950s, there were lots of researchers trying to deal with unstructured documents with computer technologies. In 1961, Doyle used an association map to describe documents. In 1967, Borko proposed an idea to index and classify documents using computer technologies [4]. In 1980, Porter [5] proposed an algorithm to remove suffixes automatically from words in

English and restore the words to its original form. In 1983, Salton recommended a vector space model (VSM) to help user compared the similarities between documents [6-9]. Salton's VSM has make progress in information retrieval from documents indexing and classifying, to comparison of similarities. Their studies have proved one thing, it's possible to deal with documents using computer technologies.

Artificial neural networks are currently attracting particular interest as methodologies to deal with unstructured data. Chen [10] applied Hopfield network to classify terms in the documents. Nigam proposed the use of maximum entropy techniques for text classification in his paper [11]. Mladenec [12] applied a naïve Bayesian classifier to select features from documents. Chen et al [13] exploited a fair feature-subset selection algorithm and an adaptive fuzzy learning network to classify online documents.

Since Kohonen introduced the SOM in the early 1980s [14], the SOM algorithm has attracted a great deal of interests among researchers in a wide variety of domains. In 1996, Lagus et al. [15] applied SOM to develop a tool called WEBSOM to cluster documents. Kohonen et al. mapped 6840568 patent abstracts onto a SOM with 1002240-node [16]. In 2002, Vicente et al. [17] exploited vector space model to extract key terms from document as input of SOM. In 2003, Smith et al. developed an LOGSOM system using Kohonen's self-organizing map to organize web pages onto a two-dimensional map [18]. In 2004, Azcarraga et al. [19] analyzed the relative weight distribution of the SOM weight vectors and took advantage of some characteristics of the random projection method to select keywords in order to extract meaningful labels each cluster on the map.

10.2.2 Self Organizing Map Algorithm

The SOM is an artificial neural network model based on unsupervised learning, introduced by Kohonen in the early 1980s. The SOM is a tool for automatically arranging high dimensional statistical data onto a two-dimensional map. Therefore, it is able to transfer sophisticated, non-linear statistical relationships among high dimensional data into simple, low dimensional map.

In our study, we adopted Smith's revised SOM algorithm [18]. The algorithm is summarized as follow.

- **Initialize input nodes, out nodes, and connection weights.** Initialize a small random value for W_{ij} as seeds. Assign N_k as neighborhood size. Assign a random value between 0 and 1 for parameter function $\alpha(t)$ and $\sigma^2(t)$. In the study, we assigned $\alpha(0) = 0.1$ and $\sigma^2(0) = 1$.

- **Calculate the distance with Euclidean distance.** Present an input vector x_i to represent document x_i and calculate the similarity d (distance) of the input vector to the weights w of each node j using Eq. 10.1.

$$d_j = \|x - w_j\| = \sqrt{\sum_i^n (x_i - w_{ij})^2} \quad (10.1)$$

- Select the node with minimum Euclidean distance as the winner node k .
- Update the weight. Applying Eq. 10.2 to adjust the weight related to the input layer and winning node, and its neighboring nodes.

$$w_{ij}(t+1) = w_{ij}(t) + c(x_i - w_{ij}(t)) \quad (10.2)$$

where $c = \alpha(t) \exp(-\|r_i - r_k\| / \sigma^2(t))$, and $r_i - r_k$ is the number of nodes, to represent the physical distance between node j and winning node k .

- **Repeat the step above, until the weights have stabilized.** In each epoch, increase t by 1, and decrease the neighborhood size, $\alpha(t)$ and $\sigma^2(t)$ by Eq. 10.3.

$$\alpha(t) = \alpha(0) N_k(t) / N_k(0) \quad (10.3)$$

N_k : size of neighborhood.

- **The final stage.** The iterative step will be terminated after all the vectors of documents have been presented to the training network, and mapped onto a two-dimensional map.

10.2.3 Users' information need

According to the researches of Baeza-Yates [20], there are three seeking techniques for most users to fulfill their information needs.

The first technique is browsing. Marchionini defined browsing behavior as “*an exploratory, information seeking strategy that depends upon serendipity and especially appropriate for ill-defined problems and for exploring new task domains*” [21].

The second technique is searching. According to definition of Sim, searching is “*an analytical process, is a planned activity with a specific goal, such as to find a particular fact*” [22].

The third technique is visualization. Visualization interfaces, such as windows, menu, icon, and dialog boxes, make the computer system more friendly and easily to use. The goal of this study is to apply SOM algorithm to cluster Chinese documents onto a two-dimensional SOM. With

this map, it can assist users' browsing and visualization needs in information retrieval.

10.3 Methodology

In our research, we regarded documents as a bag of words. We ignored the order of the words as well as any punctuations or structural information, but retained the frequencies of every word. Our research steps are as follows.

- **Acquiring documents**

We attained 892 botanical documents from Herbarium, Research Center for Biodiversity, Academia Sinica, in Taiwan (HAST) [23] and Taiwan Agricultural Chemicals and Toxic Substances Research Institute, Council of Agriculture (TACTRI/COA) [24] as our training data. The documents are all in html format, and needed to be transferred to plain text.

- **Preprocessing documents**

All tags in the botanical documents needed to be removed and transferred to plain text as training data.

- **Segmenting Chinese terms**

CKIP was a Chinese term segmentation system, developed by Institute Linguistics, Academia Sinica, Taiwan [25]. We applied the CKIP system to extract Chinese terms from sentences.

- **Extracting Chinese key terms**

We removed the stopwords from the documents, and revised the errors which segmented by CKIP with our revising system developed by Java language.

- **Calculating term weight**

Calculate the weight of every key term. We applied Eq. 10.4 to calculate the term weight.

$$w = (tf \times idf) \quad (10.4)$$

where $idf = \log\left(\frac{N}{df}\right)$, W : the term weight, tf : the term frequency, df : the term frequency shows in the document collection, idf : the inverse df .

- **Vectorizing documents**

After the term extracting and weight calculating, the 892 botanical documents will transfer into vectorized statistical data shown as Eq. 10.5.

$$D_i = (W_1, W_2, W_3, \dots, W_n) \quad (10.5)$$

where D_i represents the 892 documents, and W represents the term weight.

• **Clustering documents with SOM**

The Vectorized documents will be used as input data of SOM algorithm. After the training processes, the input data will be mapped to the Kohonen layer.

• **Projecting documents onto a SOM map**

The vectorized data $D_1, \dots, D_i$ will be projected onto a two-dimensional grid map. This map can represent the relationships among documents. The features of adjacent nodes will be similar to the centroid node.

10.4 Results

In this study, we applied Vector Space Model, developed by Salton to vectorize the 892 botanical documents into statistical data as the input value of SOM algorithm. We assigned 20x20 nodes in our research. The learning rate would be 0.1, 0.5, and 0.9; learning epochs would be 100, 200, 400, and 500.

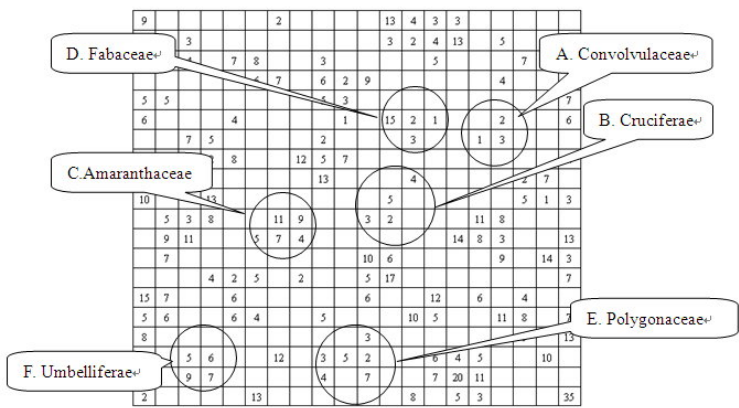


Fig. 10.1. Analysis of botanical document map

10.4.1 Discussion

Figure 10.1 represents an output map of SOM, Each number on the cell points the amount of botanical documents clustered into this cell. There are two documents in cell (10, 4). Those features of these two documents were “葉對生 (Phyllotaxy: Opposite) 長橢圓形 (Leaf shape: Oblong) 葉尖銳

(Apex: Acuminate) 全緣 (Leaf margin: Entire) 花白色 (flower: White) 球形 (Inflorescence: sphere) 腋生 (Inflorescence: axillary)". When we look up the botanical illustration, we found they are the same plant, but using different Chinese name. Both of them are "Alternanthera philoxeroides (Mart.) Griseb" in the scientific name of "長梗滿天星" and the nickname of "空心蓮子草" in Chinese. They have been clustered into the same area correctly.

10.4.2 Evaluation

To Evaluate the efficiency of an information retrieval system, Baeza-Yates [23] proposed recall rate and precision rate as the measurements. Eq. 10.6 is the formula for calculating recall rate, and Eq. 10.7 is the formula for calculating precision rate.

$$\text{Recall} = \frac{|\text{number of relevant Docs retrieved}|}{|\text{number of relevant Docs in collection}|} \quad (10.6)$$

$$\text{Precision} = \frac{|\text{number of relevant Docs retrieved}|}{|\text{number of Docs retrieved}|} \quad (10.7)$$

Figure 10.1 depicts the analysis of botanical document map, and Table 10.1 shows the result of our study.

In our experimental results, the lowest recall was 0.71 for Polygonaceae documents and the highest recall rate was 0.94 for Amaranthaceae documents. The lowest precision rate was 0.81 for Umbelliferae documents, and the highest precision rate was one hundred percent for Convolvulaceae and Cruciferae documents.

Table 10.1. Analysis of recall and precision rate

Area	Family	Related docs been retrieved	Related docs in collection	Retrieve d docs	Recall rate	Precision rate
A	Convolvulaceae	6	8	6	0.75	1
B	Cruciferae	14	15	14	0.93	1
C	Amaranthaceae	32	34	36	0.94	0.89
D	Fabaceae	19	25	21	0.76	0.90
E	Polygonaceae	22	31	24	0.71	0.92
F	Umbelliferae	22	30	27	0.73	0.81

10.5 Conclusion

In this work we applied SOM algorithm to classify the Chinese botanical documents onto a two-dimensional map. The method performed a completely automatic and unsupervised clustering of the botanical documents.

The result shown in Figure 10.1 depicted a lowest recall rate 0.71 for Polygonaceae and lowest precision rate 0.81 for Umbelliferae. The experimental results depicted a good performance in recall and precision rate. This botanical document map can assist users to browse information with visualization SOM map.

In the future research, we hope to adopt other artificial neural networks to cluster or classify unstructured data, and improve the efficiency of information retrieval.

References

1. Kohonen T, Kaski S, Lagus K, Salojärvi J, Honkela J, Paatero V, Saarela A (2000) Self-organization of a massive document collection. *IEEE Trans. Neural Networks* 11:574-585
2. Fang R-J, Lo H, Weng Y-C, Tsai HL (2007) Mobile learning system using multi-dimension data warehouse concept-based on botanical data. In: *Proceeding of the 6th WSEAS International Conference on Applied Computer Science*, Hangzhou, China
3. Sullivan D (2001) *Document warehousing and text mining*. John Wiley & Sons, Inc.
4. Trybula WJ (1999) *Text mining and knowledge discernment: An exploratory investigation*. Ph.D., The University of Texas at Austin
5. Porter M (1980) *Readings in information retrieval*. Morgan Kaufmann, San Francisco
6. Salton G, Wong A, Yang CS (1975) A vector space model for automatic indexing. *Communications of the ACM* 18:613-620
7. Salton G, McGill M (1983) *Introduction to modern information retrieval*. McGraw Hill, New York
8. Salton G, Buckley C (1988) Term-weighting approaches in automatic text retrieval. *Information Processing and Management* 24:513-532
9. Salton G, Singhal A (1995) *Automatic text browsing using vector space model*. Department of Computer Science, Cornell University
10. Chen H, Ng T (1995) An algorithmic approach to concept exploration in a large knowledge network (Automatic Thesaurus Consultation): Symbolic Branch-and-Bound Search vs. Connectionist Hopfield Net Activation. *Journal of the American Society for Information Science* 45:348-369

11. Nigam K, Lafferty J, McCallum A (1999) Using maximum entropy for text classification. In: IJCAI-99 Workshop on Machine Learning for Information Filtering, Stockholm, Sweden, pp 61-67
12. Mladenic D, Grobelnik Marko (2003) Feature selection on hierarchy of web documents. *Decision Support Systems* 35:45-87
13. Chen C-M, Lee H-M, Tan C-C (2006) An intelligent web-page classifier with fair feature-subset selection. *Engineering Applications of Artificial Intelligence* 19:967-978
14. Kohonen T (1995) *Self-organizing maps*. Springer, Berlin, Heidelberg
15. Lagus K, Honkela T, Kaski S, Kohonen Teuvo (1996) Self-organizing maps of document collections: A new approach to interactive exploration. In: *Proceedings of the Second International Conference on Knowledge Discovery & Data Mining*
16. Kohonen T, Kaski S, Lagus K, Salojärvi J, Honkela J (2000) Self organization of a massive document collection. *IEEE Transactions on Neural Networks* 11:574-585
17. Guerrero VP, Moya-Anegón FL, and Herrero-Solana V (2002) Automatic extraction of relationships between terms by means of Kohonen's algorithm. *Library & Information Science Research* 24:235-250
18. Smith KA, Ng A (2003) Web page clustering using a self-organizing map of user navigation patterns. *Decision Support Systems* 35:245-256
19. Azcarraga AP, Y Jr TN, Tan J, Chua TS (2004) Evaluating keyword selection methods for WEBSOM text archives. *IEEE Transactions on Knowledge and Data Engineering* 16:380-283
20. Baeza-Yates R, Ribeiro-Neto B (1999) *Modern information retrieval*.
21. Marchionini G, Shneiderman B (1988) Finding facts vs. browsing knowledge in hypertext systems. *IEEE Computer*, pp 70-79
22. Sim SE, Clarke CLA, Holt RC, Cox AM (1999) Browsing and searching software architectures. In: *Proceedings of the International Conference on Software Maintenance*, Oxford, England
23. T. Academia Sinica (2007) Herbarium, research center for biodiversity. Available at <http://hast.sinica.edu.tw>
24. Hsu LM, Chiang MY (2007) Lawn weeds in Taiwan. Available at <http://www.tactri.gov.tw/htdocs/plant/lawnweed/index.asp>
25. Institute Linguistic (2007) CKIP Chinese term segmentation system. Academia Sinica

Chapter 11

Linear discriminant analysis in Ottoman alphabet character recognition

Zeyneb Kurt, H. Irem Turkmen, M. Elif Karsligil

Department of Computer Engineering, Yildiz Technical University,
34349 Besiktas / Istanbul TURKEY
zeyneb@ce.yildiz.edu.tr, irem@ce.yildiz.edu.tr, elif@ce.yildiz.edu.tr

Abstract. This paper proposes a novel Linear Discriminant Analysis (LDA) based Ottoman Character Recognition system. Linear Discriminant Analysis reduces dimensionality of the data while retaining as much as possible of the variation present in the original dataset. In the proposed system, the training set consisted of 33 classes for each character of Ottoman language alphabet. First the training set images were normalized to reduce the variations in illumination and size. Then characteristic features were extracted by LDA. To apply LDA, the number of samples in train set must be larger than the features of each sample. To achieve this, Principal Component Analysis (PCA) were applied as an intermediate step. The described processes were also applied to the unknown test images. K-nearest neighborhood approach was used for classification.

Keywords. Ottoman Character Recognition, PCA, LDA

11.1 Introduction

Within the last years, rapidly growing interest in document archiving and retrieval systems increased attention to character recognition systems. Numerous approaches have been proposed for optical or handwritten character recognition mostly in Latin-based alphabets. On the contrary, some

alphabets as Arabic, Chinese, and Japanese have picture like characters which require more complex recognition algorithms to identify. This paper presents an Ottoman alphabet recognition system. Ottoman characters are the characters of an alphabet that was formed by adding some new letters from Farce and Turkish alphabet to the Arabian alphabet. Besides, Ottoman alphabet has been improved with its original features and has become an art due to its various scriptural formats.

There are few implementations of automated Ottoman alphabet recognition. Atici and Yarmen-Vural, applied chain coding to characters to propose a Hidden Markov Model based Ottoman alphabet recognition system [1]. Ozturk, A.Gunes proposed a method for recognition of isolated Ottoman characters by neural networks [2].

We propose a new global feature based approach for Ottoman alphabet recognition by using Linear Discriminant Analysis (LDA). LDA projects the data onto a lower dimensional space while preserving as much of the class discriminatory information as possible.

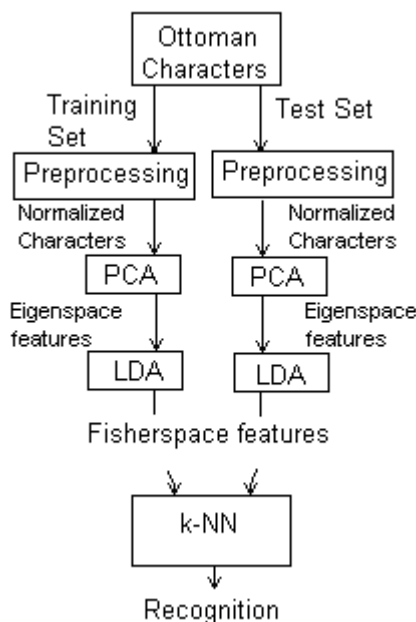


Fig. 11.1. Block diagram of proposed system

Ottoman script is fundamentally cursive. Since this paper focuses on the character recognition, the segmentation of the connected characters has been not studied. In Figure 11.1, the block diagram of the proposed system is illustrated. The outline of the paper is as follows: Section 11.2 presents

preprocessing steps of the system. Section 11.3 describes extraction of the characteristics features using LDA. In Section 11.4, decision steps of the system are described. Finally in section 11.5, experimental results of our implementation are given. Moreover, the performance of the proposed system is discussed.

11.2 The preprocessing

Ottoman characters have many common points with each other in their main shapes, whereas they differ in detail. In the proposed system, each Ottoman character is considered as a class. Our main goal is to discuss the influence of the components that points out the within class similarities and between class differences on success rate.

Since our purpose is to measure the performance on recognition of isolated characters, the classification success of separate characters is scrutinized.

The objective of LDA is to find out the most efficient combinations to split up multiple classes and to maximize between class differences. Thus, it is widely used in several applications such as face recognition, speech recognition, image summarization and data classification [5].

As a precondition of LDA implementation, the size of each sample must be less than the number of samples in the training set. Since the training set is inconvenient with that precondition, dimension reduction should be applied. For this purpose PCA is applied to training set as an intermediate step.

The main purpose of preprocessing is to reduce the variation in size and illumination. The preprocessing steps are applied both to the training and testing images.

In this study, 256 gray level images with a light colored background were used. By the contrast enhancement, not only the identification of characters was improved and the characters were slimed down, but also the potential noises in the images were eliminated. Moreover, the images were aligned and resized.

Figure 11.2, shows the preprocessing steps applied on images. For each image, the inside of the frame which was drawn by using the tangents to the maximum and minimum points of characters on x and y coordinates, is taken into consideration.

After the alignment process, images may be in several sizes. In this study, each character was normalized to 32x32 pixels.

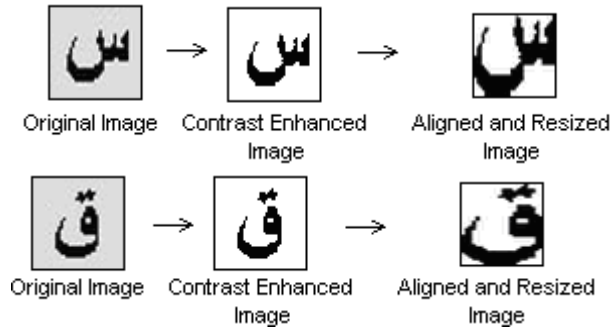


Fig. 11.2. Examples of the Preprocessing Steps

11.3 The feature extraction

Each Ottoman character has some features that are similar to others and different from others. In this study LDA was used to extract the characteristic features of Ottoman characters. To implement the LDA, the number of samples in train set must be larger than the features of each example. To achieve this, first we applied PCA. The selection of eigenvectors with the highest eigenvalues reduces the dimensionality. Then we applied LDA to this reduced feature set.

11.3.1 Principle components analysis

PCA reduces dimensionality of data while expressing most important characteristic features. PCA finds basis vectors for a subspace, which:

- maximizes the variance retained in the projected data
- or (equivalently) gives uncorrelated projected distributions
- or (equivalently) minimizes the least square reconstruction error[3].

Each of the p number of images whose size is $n*m$ is assumed to be $[n*m]*1$ sized column vectors that correspond to the original data [4].

$$\Omega = X * X^T \quad (11.1)$$

The covariance matrix (Ω) is defined by the Equation-1 where X is the mean matrix subtracted vector whose size is $[n*m]*p$, where there are p image in the training set. Because of the multidimensionality, (Ω) is calculated using (11.2), instead of (11.1).

$$\Omega = X^T * X \quad (11.2)$$

The eigenvalues and associated eigenvectors are obtained as in (11.3) where Ω is the covariance matrix, λ is the set of eigenvalues and V is the associated eigenvectors.

$$\Omega * V = \lambda * V \quad (11.3)$$

We do not have to use the whole acquired eigenvectors. Therefore the eigenvectors are sorted by descending order and only selected number of highest eigenvalues are used.

PCA was applied to both training data and test data. The acquired projections in the eigenspace were given to LDA as input.

11.3.2 Linear discriminant analysis

LDA aims to determine the linear combinations of the features which can separate the objects and events into more than one class. These combinations, which were obtained after applying LDA, can be used as a linear classifier or it can be used to decrease the dimension for the classification step.

LDA maximizes between-class scatter while minimizing the within-class scatter [6].

The projections in the eigenspace, which were obtained by PCA and belonged to the alphabet's each character, were vectorized.

The mean vector (M) of the alphabet's character set was calculated.

The mean vectors of each character class (M_i) were calculated and for each class the mean vector of the class was subtracted from each character in a class. Let m be the number of the samples in each class, n be the number of classes and A be the mean-centered eigenspace projections.

The scatter matrices for each class were acquired by:

$$S_1 = A_{11} * A_{11}^T + \dots + A_{1m} * A_{1m}^T \quad (11.4)$$

$$S_n = A_{n1} * A_{n1}^T + \dots + A_{nm} * A_{nm}^T \quad (11.5)$$

The within class scatter matrix S_w was obtained by (11.6) where M_i is the mean vector of i^{th} class and M is the mean vector of whole characters in the train set.

$$S_w = S_1 + S_2 + \dots + S_n \quad (11.6)$$

The between class scatter matrix S_B was built by:

$$S_B = 2 * \Sigma (M_i - M) \quad (11.7)$$

The eigenvectors (V) and eigenvalues (λ) were calculated as:

$$S_B * V = \lambda * S_W * V \quad (11.8)$$

The eigenvectors were sorted by their associated eigenvalues in descending order and the first $n-1$ eigenvectors were kept. Fisher space projections of each class in the train set were obtained by using these eigenvectors.

11.4 Classification

Each test image was first projected into eigenspace and then into Fisher space. The k-Nearest Neighborhood was used for classification of unknown test images. The system performance was evaluated by applying 10-fold cross validations [7].

11.5 Experimental results

We conducted experiments using a database of 10 sample images of 33 Ottoman alphabet characters. The samples, gathered from electronic resources were both handwritten and in press printed format. In k-nearest neighborhood, k was selected as 3 by 10-fold cross validation.

Figure 11.3(a) illustrates true classification of three test examples.



Fig. 11.3. (a). Three samples for the test data which were recognized correctly. **(b).** The train set samples of these characters respectively.

Figure 11.4 illustrates false classification of three test examples. As it can clearly be seen, the test examples are very similar to the characters of the incorrect classes.



Fig. 11.4. (a). Three samples for the test data which were recognized incorrectly. **(b).** Train set samples for the decision classes of the system for incorrectly recognized letters

As given in Table 11.1 success ratio of our Ottoman Character Recognition system is 88%.

Table 11.1. The success ratio of the processed system

True Classification	290
False Classification	40
Total Number of Characters	330
Recognition Ratio	88%

11.6 Conclusion

This study presents a Linear Discriminant Analysis based automatic Ottoman Alphabet Character Recognition System. This approach retains class separability while reducing the dimensionality. As the performance of our proposed system is very promising for the recognition of individual characters, in the future work we will study on a system for the recognition of cursive characters in scripts. This will yield to processing and recognition of a large number of historical documents and we will be able to archive these documents.

References

1. Atici AA, Yarman-Vural FT (1997) A heuristic algorithm for optical character recognition of Arabic script. Signal Processing 62(1):87-99

2. Ozturk A, Gunes S, Ozbay Y (2000) Multifont Ottoman character recognition. In: Proceedings of the 7th IEEE Int. Conf. on Electronics Circuits and Systems (ICECS), Jounieh, Lebanon, pp 945-949
3. Turk M, Pentland A (1991) Eigenfaces for recognition. *J. Cog. Neurosci.* 3(1)
4. Turk M, Pentland A (1991) Face recognition using eigenfaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Maui, Hawaii, USA, pp 586-591
5. Belhumeur PN, Hespanha JP, Kriegman DJ (1996) Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. In: Proc. of the 4th European Conference on Computer Vision, ECCV'96, Cambridge, UK, pp 45-58
6. Etemad K, Chellappa R (1997) Discriminant analysis for recognition of human face images. *Journal of Optical Society of America A*, pp 1724-1733
7. Kung SY, Mak MW, Lin SH (2004) Biometric authentication: A machine learning approach. Prentice Hall

Chapter 12

User requirements engineering and management in software development

Zeljko Panian

Faculty of Economics and Business, University of Zagreb, Croatia

Abstract. Requirements engineering is an often underutilized discipline in software development. Yet the economic benefits – lowered development costs, faster development and better products that delight customers – have plainly been demonstrated in study after study. In this paper, the case is made for requirements engineering as an integral part of any product lifecycle. Metrics are provided to show the positive effect of requirements engineering and requirements management (RM), from individual developers to the business to the customer. A software product is examined as an example of how to bring effective requirements management into product development.

Keywords. Information technology, software engineering, user requirements, requirements engineering, requirements management

12.1 Introduction

Many software problems arise from shortcomings in the ways that people acquire, document, agree on, and modify the product's requirements. Typical problem areas include informal information gathering, implied functionality, inadequately defined requirements, and a casual change process.

Most people wouldn't ask a construction contractor to build a custom multi-hundred-thousand-dollar house without extensively discussing their

needs and desires and refining the details progressively. Homebuyers also understand that making changes carries a price tag.

However, people blithely gloss over the corresponding issues when it comes to software development. One of the widely quoted CHAOS reports from The Standish Group relates the consequences of casual approaches to requirements engineering (The Standish Group 2005). Year after year, lack of user input, incomplete and changing requirements are the major reasons why so many information technology projects fail to deliver all of their planned functionality on schedule and within budget.

Nor do requirements issues only plague the early stages of a software project. One study at a large defence contractor found that 54 percent of all the errors were discovered *after* unit testing was complete, and that 45 percent of these were requirements or design errors (Davis 1993). Other studies also indicate that 40 to 60 percent of all defects found in a software project can be traced back to errors made during the requirements stage.

Nonetheless, many organizations still practice ineffective methods for this essential project activity. The typical outcome is an expectation gap, a difference between what developers think they are supposed to build and what customers really need.

12.2 Stakeholders' interests

Nowhere more than in the requirements process do the interests of all the stakeholders in a software or system project intersect. These stakeholders include end users, customers acquiring the product, requirements analysts, project managers, developers, testers, regulators and auditors, manufacturing staff, sales and marketing, and field support or help desk staff.

Handled well, this intersection can lead to exciting products, delighted customers, and fulfilled developers. Handled poorly, it is the source of misunderstanding, frustration, and friction that undermine the product's quality and business value.

Requirements are the foundation for all the subsequent software development and project management activities. Therefore, all stakeholders must be committed to following an effective requirements process. This process includes activities for:

- Requirements development (eliciting, analyzing, specifying, and validating requirements).
- Requirements engineering (dealing with the requirements once you have them in hand).

Developing and engineering requirements is hard! There are no simple shortcuts or magic solutions. But a toolkit of techniques, coupled with tools to store and manage requirements, provides a powerful defence against the expectation gap.

12.3 Requirements are interrelated

If John Donne had been a business analyst rather than a 17th century poet, he might have said, “No requirement is an island.” Requirements have a natural parent-child relationship because high-level user requirements can be decomposed into lower level functional requirements that, in turn, generate specific test requirements.

For example, the requirement ‘Generate e-mail confirmation to customer online order’ would produce a number of distinct functional requirements (‘Verify e-mail address,’ ‘Transmit message,’ ‘Receive returned e-mail message,’ etc.), each of which needs to be implemented in a code module and requires testing.

Understanding these relationships is essential to planning and executing adequate test coverage. And when analyzing a proposed change to a requirement – or test procedure – it should be known what other requirements may be affected by the change.

Traceability between requirements has to be maintained since this provides a useful tool for the analyst or software designer when trying to assess the software product quality. For example:

- Users can define multiple types of requirements and maintain relationships between them.
- Requirement types can be customized for each project.
- Trace-to and trace-from views can display the parent/child relationship between requirements.
- Requirements can be organized into folders for easier navigation.
- Stakeholders need to be automatically informed via e-mail or requirements traceability alerts when a requirement changes.
- Audit trail can track which values are changed and when.

But, traceability between requirements only solves part of the problem. It is also essential to follow individual requirements through the application lifecycle to test, defects and release. That means testers and quality managers should be able see what and how much testing is planned for each requirement. They can then analyze the status of the testing to see how much has been completed and how many defects have been found.

And when defects are found, it lets them see which requirements are affected.

12.4 The need for requirements management

The major consequence of requirements problems is rework – doing something again that you thought was already done. Rework can consume 30 to 50 percent of total development cost (Boehm and Papaccio 1988) and requirements errors account for 70 to 85 percent of the rework cost (Lefingwell 1997). As illustrated in Figure 12.1 it costs far more to correct a defect that is found late in the project than to fix it shortly after it was created (Grady 1999).

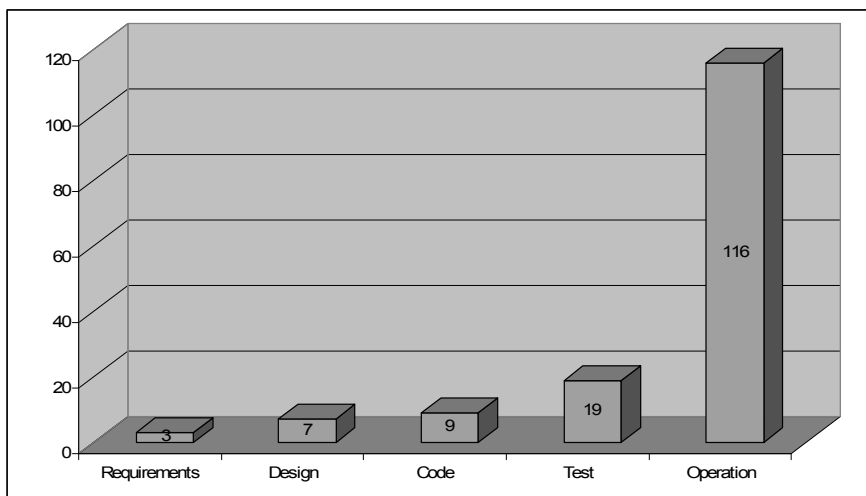


Fig. 12.1. Relative cost to correct a requirement defect depending on when it is discovered

Preventing requirements errors and catching them early, therefore, has a huge leveraging effect on reducing rework.

Shortcomings in requirements cause no end of headaches on software projects. For example, creeping requirements contribute to schedule and budget overruns. Now, change is going to happen. Requirements change because initial elicitation activities are imperfect, because business needs evolve, and because customer expectations change once they see the product beginning to take shape.

Project plans and commitments therefore need to anticipate change, rather than expecting that an initial stage of gathering requirements is all

that's needed. Requirements creep, though, is the uncontrolled growth of requirements well beyond the original scope boundary.

The way in which requirements changes are incorporated into the product can degrade quality. Changes need to be introduced at the highest level of abstraction they affect and follow the links in the traceability chain to see the impact of each proposed change. Traceability data stored in an appropriate requirements management tool greatly facilitates this impact analysis.

Many projects suffer from gold-plating. Developers might add unnecessary features they think the users will like, or users request excessively complex systems. Tracing each functional requirement back to its origin – such as a user task, a business rule, or a business objective – helps solve this problem.

The result of all of these kinds of requirements problems is to slow the pace of software development. It doesn't do you any good to work very efficiently on the wrong requirements. If you don't get the requirements right, it doesn't matter how well your team executes the rest of the project.

12.5 Accelerating development

A sign in a high-school chemistry lab reads, "If you don't have time to do it right, when will you have time to do it over?" (Bates 2003).

To convince itself of the benefits, a firm should think about how much effort on their projects is misspent because of miscommunications, misunderstandings about requirements, missing information, or misstated requirements. The case for emphasizing solid requirements practices is an *economic argument*, not a philosophical or technical argument.

A solid foundation of agreed-upon business requirements and project objectives establishes a shared vision, goals, and expectations. Frequent and intimate user involvement makes software development a collaborative partnership between IT and the customer community. It also reduces the chance of system rejection upon rollout.

It is frustrating for everyone to release a product only to find that essential functionality is missing and other capabilities are included that no one will ever use. Good requirements processes ensure that the functionality built enables users to perform essential business tasks. Well-specified requirements also define achievable quality expectations, letting the team implement both the capabilities and the characteristics that will make the users happy.

While typical projects spend perhaps 10 percent of their effort on requirements, investing more has a big payoff. A study of 15 projects in the telecommunications and banking industries revealed that the most successful projects spent 28 percent of their resources on requirements engineering (Hofmann and Lehner 2001).

The average project devoted 15.7 percent of its effort and 38.6 percent of its schedule to requirements engineering. NASA projects that invested more than 10 percent of their total resources on requirements development had substantially lower cost and schedule overruns than projects that devoted less effort to requirements (Hooks and Farry 2001).

In a European study, teams that developed products more quickly devoted more of their schedule and effort to requirements than did slower teams, as shown in Table 12.1.

Table 12.1. Investing in Requirements (Blackburn et al. 1996)

	Effort Devoted to Requirements	Schedule Devoted to Requirements
Faster Projects	14%	17%
Slower Projects	7%	9%

Not all of the requirements development effort should be allocated to the beginning of the project. Don't think of it as a "requirements phase," but rather a set of requirements-development activities that are threaded throughout the project's lifecycle. Projects that follow an incremental lifecycle will work on requirements in every iteration through the development process.

Agile development projects take an incremental approach, rapidly building small portions of the product so users can refine their needs and developers can keep up with changing business demands. Agile projects will have frequent but small requirements development efforts (Light 2005). Regardless of how large an increment your team tackles, they need to understand the requirements for that increment.

12.6 Requirements development and management

The goal of requirements *development* is to deduce, capture, and agree upon a set of functional requirements and product characteristics that will achieve the stated business objectives. Requirements *management*, then, involves everything that happens with the requirements in the product lifecycle once the requirements are developed, as shown in Figure 12.2.

The demarcation between requirements development and requirements management is the defining of a requirements baseline. A baseline is a snapshot in time that represents the agreed-upon set of requirements for a specific product release.

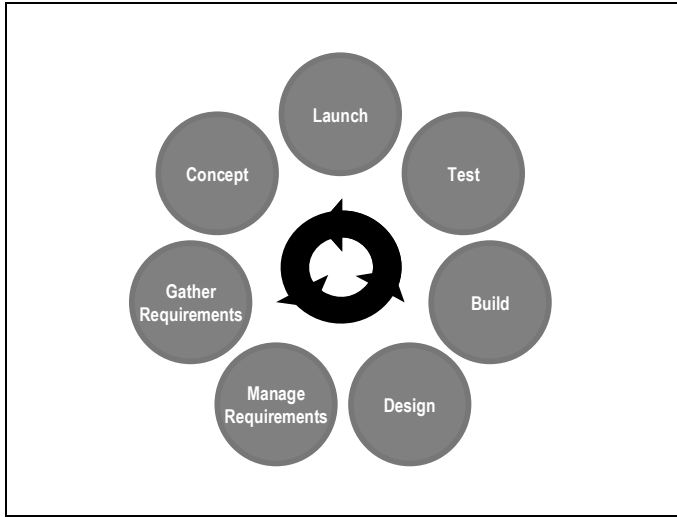


Fig. 12.2. Product Life Cycle

12.7 Strategies and best practices for better user requirements management

A variety of strategies and practices can help software development teams bridge communication gaps and do a better job of understanding, documenting and communicating customer needs. Figure 12.3 synthesizes several best practices we evidenced in our research, and yet in the categories of elicitation, analysis, specification, validation, and management.

- *Requirements Elicitation* – Defining the product vision and project scope; identifying stakeholders, customers and users; choosing elicitation techniques; exploring user scenarios.
- *Requirements Analysis* – Creating visual analysis models; building and evaluating prototypes; prioritizing requirements using previously selected attributes.
- *Requirements Specification* – Looking for ambiguities in text-based requirements; storing requirements in a central, usually shared database

for better communication; tracing requirements into design, code and tests.

- *Requirements Validation* – Reviewing the requirements; creating test cases from requirements.
- *Requirements Management* – Managing requirements versions; adopting a change control process; performing requirements change impact analysis; storing requirements attributes; tracking the status of each requirement.

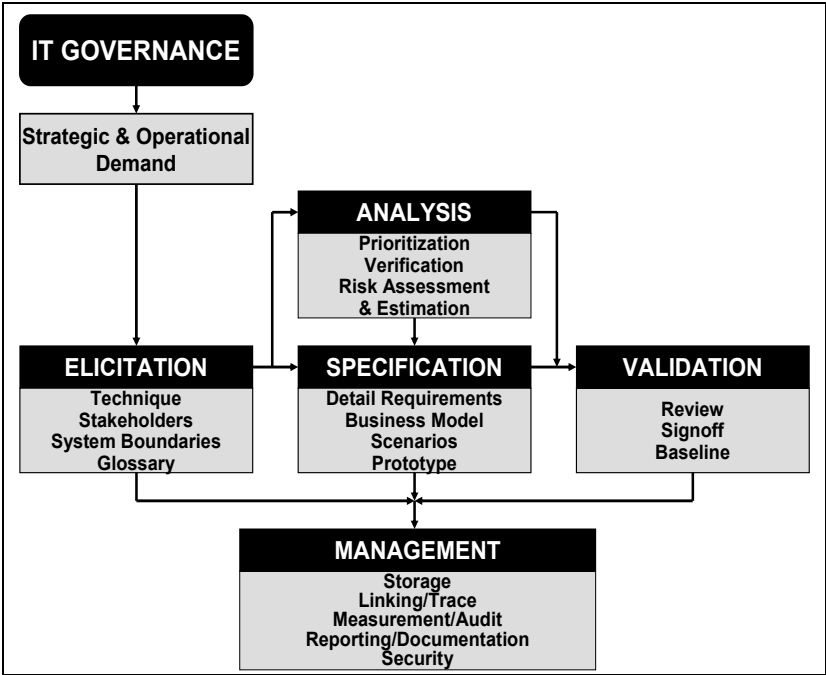


Fig. 12.3. Strategies for Requirements Management

12.8 Traditional vs. novel approach to RM

The traditional requirements engineering and management approach is to create documents that contain business, user, functional, and non-functional requirements written in natural language, preferably using the organization's standard templates.

But, it appears that the traditional document-based approach to storing requirements has numerous limitations (Ellis 2006):

- It is difficult to keep the documents current and synchronized.
- Communicating changes to all affected team members is usually a manual process.
- It is not easy to store supplementary information (attributes) about each requirement.
- It is hard to define logical links between requirements of various types and other system elements.
- Tracking the status of individual requirements is cumbersome.
- Concurrently managing sets of requirements that are planned for different releases or for related products is difficult. When a requirement is deferred from one release to another, an analyst needs to move it from one software requirements specification (SRS) to the other.
- There is no convenient place to store those requirements that were proposed but rejected, nor those requirements that were deleted from a baseline. Sometimes someone may want to remember that these requirements had been proposed, because they might be back in scope one day.
- Reusing a requirement means that the analyst must copy the text from the original SRS into the SRS for each other system or product where the requirement is to be used.
- It is difficult for multiple project participants to modify the requirements, particularly if they are geographically separated.

A novel approach to requirements management suggests usage of a software tool that can store information in a multi-user database, thus providing a robust solution to these restrictions. Users should be able to create various classes of requirements information and define unique sets of attribute values for each requirement class.

Users should import requirements from source documents, filter and display the database contents, and export requirements in various formats. They should also define traceability links between pairs of objects stored in the database, as well as connecting requirements to items stored in other software development tools.

Following are some of the tasks such a tool should help users to perform:

- *Tracking requirements status* – Collecting the requirements into a database lets user know how many discrete requirements he has specified for the product. Tracking the status of each requirement during development supports the overall status tracking of the project. A project manager has good insight into project status if he or she knows that, e.g., 55 percent of the requirements committed to the next release have

been verified, 28 percent have been implemented but not verified, and 17 percent are not yet fully implemented.

- *Communicate with stakeholders* – The tool should permit team members to discuss requirements issues electronically through threaded and interactive conversations. Additionally, e-mail messages can automatically be triggered to notify affected individuals when a specific requirement is modified or a change is proposed. Making the requirements accessible online can save travel costs and reduce document proliferation.
- *Store requirements attributes* – Possible attributes that provide a richer understanding of each requirement include priority, status, verification method, origin, person responsible, planned release, and so on. Everyone working on the project must be able to view the attributes and selected individuals be permitted to update their values (Frankl and King 2005). The tool should generate system-defined attributes such as the date a requirement was created and its current version number, and it lets you define additional attributes of various data types. One can sort, filter, or query the database to display subsets of the requirements that have specific attribute values. For example, a user might ask to see a list of all the requirements planned for release 2.3 that were assigned to Julie and have a current status of Implemented.
- *Manage versions and changes* – The project should define a requirements baseline. Software tool should provide flexible baselining functions and maintains a history of the changes made to every requirement. You can record the rationale behind each change decision and revert to a previous version of a requirement if necessary. The tool should also contain a built-in change-proposal system that links change requests directly to the affected requirements.
- *Facilitate impact analysis* – The appropriate software tool should enable requirements tracing by letting user define links between different kinds of requirements, between requirements in different subsystems, and between individual requirements and related system components – for example, designs, code, tests, and user documentation. These links then help when analyzing the impact that a proposed change will have on a specific requirement by identifying other system elements the change might affect. It is also a good idea to trace each functional requirement back to its origin or parent so that you know where every requirement came from.
- *Control access* – An organization can define access permissions for individuals or groups of users. The software tool should enable information sharing with a geographically dispersed team through a Web inter-

face to the database. The database uses requirement-level locking to permit multiple users to update the database contents concurrently.

- *Reuse requirements* – Storing requirements in the tool's database facilitates reusing them in multiple projects or subprojects. Requirements that logically fit into multiple parts of the product description can be stored once and referenced when necessary to avoid duplicating requirements.

In short, the software tool should be a powerful complement to a robust requirements engineering process. The tool alone will not guarantee that high-quality requirements are written, that system designer has talked with the right user representatives or that he has set the right priorities. However, the tool can keep a project team aligned on the specified requirements for each release, so they all agree when they are done and there will not be so many surprises when they get there.

12.9 Assessing return on investment

Our research has convinced us that estimating the payback that better requirements can bring to the organization requires considering the following questions:

- What fraction of organization's own development effort is expended on rework (due to miscommunication)?
- How much does a typical customer-reported defect cost the organization?
- How much costs them a defect found in system testing?
- What fraction of user-reported defects and what fraction of defects discovered during system testing originate in requirements errors?
- How much of the organization's costs – such as defect correction and unplanned enhancements – can be attributed to missing requirements or other requirements inadequacies?

Aside from obvious benefits, improving requirements approaches leads to other valuable – though less tangible – outcomes. Fewer miscommunications on a software project reduce the overall level of disorder. Less disorder lowers unpaid overtime, increases team morale, boosts employee retention and improves the team's chances of delivering on time.

Best of all, these benefits ultimately lead to higher customer satisfaction.

12.10 Conclusion

Software development involves roughly 50 percent computing and 50 percent communication. Unfortunately, most IT professionals and teams are better at computing – user requirements, however, are almost entirely about communication.

Because there are many links in the requirements communication chain, a breakdown in any of these links leads to significant problems. For example, if an analyst misunderstands stakeholder input about requirements, if important requirements information does not surface, or if an analyst and developer do not share the same understanding about requirements, the resulting product fails to meet customers' needs.

The inevitable outcome of requirements errors is time-consuming and costly rework. Experts report that rework can consume 30 to 40 percent of the total effort expended on a software project. Our study has indicated that nearly 50 percent of the defects identified on software projects observed can be traced back to errors in the requirements.

Our analysis of potential return on investment from requirements suggests that requirements errors can consume between 70 and 85 percent of all project rework costs. It can cost up to 110 times more to correct a requirements defect found in operation than it would if that same defect has been discovered during requirements assessment and definition.

To convince itself of the benefits, a firm should think about how much effort on their projects is misspent because of miscommunications, misunderstandings about requirements, missing information, or misstated requirements.

Requirement engineering and management is a developing discipline that tries to solve those problems. A novel approach to requirements engineering and management suggests usage of a software toolkit that can store information in a multi-user database, thus providing a robust solution to these restrictions.

References

1. Bates J (2005) Business in real time – Realizing the vision. *DM Review*, 05/2003 (<http://www.dmreview.com/>)
2. Blackburn JD, Scudder GD, Van Wassenhove LN (1996) Improving speed and productivity of software development: A global survey of software developers. *IEEE Transactions on Software Engineering* 22(12):875-885
3. Boehm BW, Papaccio PN (1998) Understanding and controlling software costs. *IEEE Transactions on Software Engineering* 14(10):1462-1476

4. Davis AM (1993) *Software requirements: Objects, functions, and states*. Prentice Hall PTR, Upper Saddle River (NJ)
5. Ellis K (2006) The executive briefing on the issues surrounding getting business requirements right. <http://www.digital-mosaic.com>, May 2
6. Frankl A, King T (2005) How to detect requirements errors – A guide to slashing software costs and shortening development time. <http://www.n8systems.com>, July 22
7. Grady RB (1999) An economic release decision model: insights into software project management. In: *Proceedings of the Applications of Software Measurement Conference*, Orange Park (FL), pp 227–239
8. Hofmann HF, Lehner F (2001) Requirements engineering as a success factor in software projects. *IEEE Software* 18(4):58–66
9. Hooks IF, Farry KA (2001) Customer-centered products: Creating successful products through smart requirements management. In: *Proceedings of the AMACOM Conference*, Miami (FL), pp 207–215
10. Leffingwell D (1997) Calculating the return on investment from more effective requirements management. *American Programmer* 10(4):13–16
11. Light M (2005) Agile requirements definition and management will benefit application development. <http://www.gartner.com>, Apr. 18
12. The Standish Group Staff (2005) *The CHAOS report on software design*. The Standish Group, Palo Alto (CAL)

Chapter 13

SSAF – A tool for active filter synthesis

Elena Doicaru,¹ Lucian Bărbulescu,¹ Claudius Dan²

¹Faculty of Automation, Computer Science and Electronics, University of Craiova str. Decebal, Nr.5, 200646 Craiova ROMANIA,
dmilena@electronics.ucv.ro, luci@cs.ucv.ro

²Faculty of Electronics, Telecommunication and Information Technologies, POLITEHNICA University of Bucharest, blvd. Iuliu Maniu, Nr. 1-3, 77209 Bucharest, ROMANIA, claus@mESsnet.pub.ro

Abstract. In this paper a software tool used to determine the topological elements of active analog continuous-time filters is presented. The filter is described by an arbitrary transfer functions expressed as $t(s) = p(s) / e(s)$, where $e(s)$ is the natural mode (pole) polynomial and $p(s)$ is the transmission zero polynomial. The goal is to obtain filters having small sensitivities, low noise and high dynamic ranges. The synthesis method implemented in this program is based on the intermediate transfer functions (IFs) method.

Keywords. Synthesis, Active Filters, Low Sensitivities, and Low Noise

13.1 Introduction

In the VLSI era, as mixed analog/digital circuits of increasing complexity are implemented into a single IC, the need for low-sensitivity, low-noise, high performance filters steadily increases. A major application for continuous-time filters is direct signal processing, especially for medium dynamic range applications, when high speed and/or low power operation are needed. For this reason we developed a software tool for automate synthe-

sis of analog continuous-time filters – SSAF (Automatic Structural Synthesis of Active Analog Continuous-Time High-order Filters). The software tool was developed in C++ using Visual C++ 2005 programming environment and can be used to determine the topological elements of active analog continuous-time filters, described by an arbitrary transfer function in the form $t(s) = p(s)/e(s)$, where $e(s)$ is the natural mode (pole) polynomial (of order n) and $p(s)$ is the transmission zero polynomial (of order m). The goal is to obtain small sensitivities, very low noise and good dynamic range.

The synthesis method selected is the intermediate transfer functions (IF's) method (Snelgrove and Sedra 1986). The method consists in obtaining an n -th transfer function, $t(s)$, by means of a structure of n resistively interconnected integrators using the linearly independent IFs selected from the given $t(s)$ following the general structure presented in Figure 13.1. The IFs have identical denominator polynomials, $e(s)$, and arbitrary numerator polynomials of degree less than n . The program selects the IF's that are able to provide optimum performance structures. The significant advantage of this method consists in the possibility to optimize the filter performance at the abstract level of IFs while using other methods only topological level optimization is possible.

13.2 Program's structure and algorithm

The filter structure presented in Figure 13.1 can be described by a state-variable formulation:

$$s \cdot \mathbf{x}(s) = \mathbf{A} \cdot \mathbf{x}(s) + \mathbf{b} \cdot u(s) + \boldsymbol{\varepsilon}(s), \quad y(s) = \mathbf{c}^T \cdot \mathbf{x}(s) + d \cdot u(s) \quad (13.1)$$

In equation (13.1) vector $\mathbf{x}(s)$ represents the circuit state (integrators output voltages), matrix $\mathbf{A}$ describes the interconnections between the n integrators, vector $\mathbf{b}$ contains the coefficients that multiply the input signal $u(s)$, vector $\mathbf{c}$ contains the coefficients required to form the output, scalar d is the coefficient of the feed-through component and $\boldsymbol{\varepsilon}(s)$ is the vector containing the noise components at the integrator inputs.

The dual sets of IFs, $\{f_i(s)\}$ and $\{g_i(s)\}$, are given by:

$$\begin{aligned} f_i(s) &= x_i(s)/u(s); \quad \mathbf{f}(s) = (s \cdot \mathbf{I} - \mathbf{A})^{-1} \cdot \mathbf{b} \\ g_i(s) &= y(s)/\varepsilon_i(s); \quad \mathbf{g}^T(s) = \mathbf{c}^T \cdot (s \cdot \mathbf{I} - \mathbf{A})^{-1} \end{aligned} \quad (13.2)$$

The $\{f_i(s)\}$ set contains the transfer function from the filter input to the integrator outputs and the $\{g_i(s)\}$ set can be physically interpreted as the integrators noise gains.

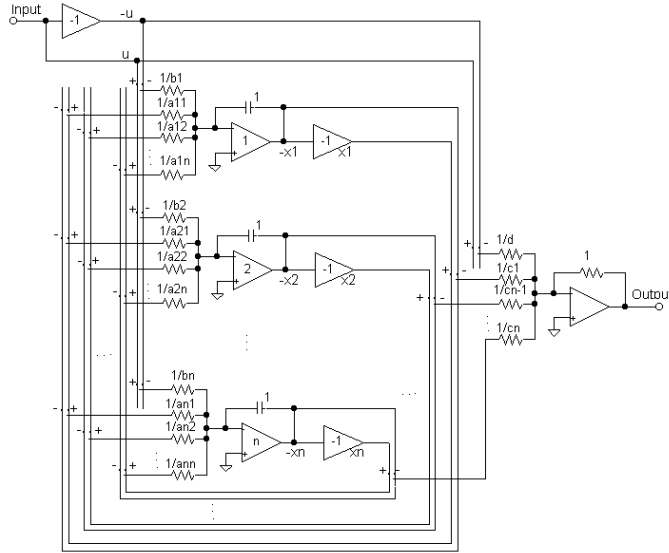


Fig. 13.1. Generic Filter Structure

From these sets we can obtain the $\{\mathbf{A}, \mathbf{b}, \mathbf{c}, d\}$ topological parameters using the following relationships:

$$\mathbf{A} = \mathbf{F} \cdot \mathbf{E} \cdot \mathbf{F}^{-1}, \mathbf{b} = \mathbf{F} \cdot \mathbf{1}, \mathbf{c}^T = \mathbf{t}^T \cdot \mathbf{F}^{-1}, d = t_{n+1}, t(s) = \mathbf{t}^T \cdot \mathbf{v}(s) + t_{n+1} \quad (13.3)$$

$$\mathbf{f}(s) = \mathbf{F} \cdot \mathbf{v}(s), \mathbf{g}(s) = \mathbf{G} \cdot \mathbf{v}(s), v_i(s) = 1/(s - e_i), i = 1, n, \mathbf{G}^T = \mathbf{H} \cdot \mathbf{F}^{-1}$$

where $\mathbf{t}$ is a vector containing the n residues of $t(s)$ at the poles, t_{n+1} is the residue at $s = \infty$, $\mathbf{F}$ is a matrix containing the residues of the f functions at the poles, $\mathbf{G}$ is a matrix of the residues of the g functions, e_i are the roots of $e(s)$, $\mathbf{E}$ is the diagonal matrix having the roots e_i as its elements and $\mathbf{H}$ is a diagonal matrix formed from the residues of $t(s)$. The sensitivities of the $\mathbf{A}, \mathbf{b}, \mathbf{c}, d$ parameters, the integrator gain γ_i and the operational amplifier gain $\mu_i(s)$ hardly depend by the IFs set (Snelgrove and Sedra 1986). In the terms of $\{\mathbf{A}, \mathbf{b}, \mathbf{c}, d\}$ elements the gain of the integrator i is proportional to the i row sum of system (13.1.a). From classical sensitivities results (Snelgrove and Sedra 1980) can be demonstrated that the minimum sensitivity realization has:

$$f_i(s) \cdot g_i(s) = -(1/n) \cdot [dt(s)/ds], \forall i = \overline{1, n} \quad (13.4)$$

If we assume that the noise signals, modeled by $\mathbf{e}(s)$, have white spectra with equal densities N_i^2 , the output noise will have a power spectrum and a rms level given by:

$$P_{n0}(\omega) = N_i^2 \cdot \sum_i |g_i(j \cdot \omega)|^2, \|P_{n0}(\omega)\| = N_i^2 \cdot \sqrt{\sum_i \|g_i(j \cdot \omega)\|_2^2} \quad (13.5)$$

The mathematical analysis emphasizes the fact that the best realizations are those obtained by orthogonal, orthonormal and derivative IFs sets. The SSAF program generates all these types of IFs sets. The general structure of the program is presented in Figure 13.2.

An IFs set is orthogonal if the all functions of the set are linearly independent and their inner product is zero. An IFs set is orthonormal if all $f_i(s)$ are orthogonal and $\|f_i\|_2 = 1, \forall i = 1, n$. Such sets can be obtained from any set of independent $\{u_i(s)\}$ by using the Gram-Schmidt ortho-normalization procedure (Şabac 1981). The inner products, $u_i \bullet v_j$, and the norm of functions, $\|v_j\|_2 = \sqrt{v_i \bullet v_i}$, required by the Gram-Schmidt procedure, are determined using the residue theorem. This is possible because of rational form of functions. So the program uses the following relations:

$$u_i \bullet v_j = \int_{-\infty}^{\infty} R_{ij}(\omega) \cdot d\omega = 2 \cdot \pi \cdot j \cdot \sum_{\substack{m \\ \text{Im}(pol_m) > 0}} \text{rez}(R_{ij}, pol_m) \quad (13.6)$$

The function residues are determined in conformity with classical definition. The orthonormal functions are $f_i = v_i / \|v_i\|_2$.

Then the topological elements are determined in conformity with relation (13.3). Their sensitivities and the filter noise are evaluated and the following sensitivity and noise criteria are listed:

$$SA = \max_{i,j, \omega \in R_a} \text{Real}(S_{Aij}^{t(s)}), Sb = \max_{i, \omega \in R_a} \text{Real}(S_{b_i}^{t(s)}), Sc = \max_{i, \omega \in R_a} \text{Real}(S_{c_i}^{t(s)}), \quad (13.7)$$

$$Sd = \max_{\omega \in R_a} \text{Real}(S_d^{t(s)}), Sd = \max_{\omega \in R_a} \text{Real}(S_d^{t(s)}), S\gamma = \max_{i, \omega \in R_a} \text{Real}(S_{\gamma_i}^{t(s)})$$

$$P_{zg} = \|P_{n0}(\omega)\|_2 \big|_{N_i=1} = \sqrt{\sum_i \|g_i(j \cdot \omega)\|_2^2}$$

Since sensitivities are proportional to $\{f_i(s)\}$ it results that various sensitivities (magnitudes and phases) go to zero in the pass-band. In a ladder only magnitude sensitivity is zero.

From eq. (13.4) one can see that if the minimum sensitivity realization exists, then $\{f_i(s)\}$ must divide $t'(s)$. In the SSAF program, $\{f_i\}$ are chosen so

that the factors of $t'(s)$ missing from $f_i(s)$ (that could be the factors of $g_i(s)$ in the case of minimum-sensitivity realizations) can appear in f_{n-i} (Doicaru 1998).

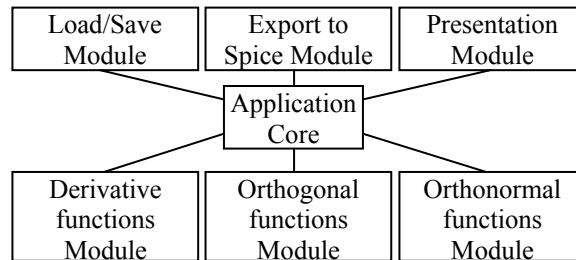


Fig. 13.2. Application Structure

Unlike orthogonal and orthonormal IF's that provide a single set of n functions f_i , in the DIF's case we can obtain a lot of n functions sets. The program generates all DIF's sets and selects one or more DIF's sets, following the user options. The first step in the DIF's sets determination is to obtain the derived transfer function $t'(s)$ and his roots. The DIF's can't be generated in two cases: (1) if the transmission zero polynomial order of $t'(s)$ is less than n ; (2) if in all sets of IF's we have at least a function with the transmission zero polynomial order greater than n . The next step is to separate the real roots from the complex roots. The conjugated complex roots are grouped in binomials. Then, the monomials and binomials are binary coded. This binary code facilitates the simultaneous generation of f_i and f_{n-I} DIF's of a set. The number of monomials and binomials possible combinations is reduced by generating only the combinations that yield polynomials with order less than n . These combination variants, representing all possible physical variants, are combined once again in $n/2$ groups.

13.3 SSAF program - Implementation and operation

As presented previously, the application contains several modules, each with its own role, as follows:

- *Load/Save Module*: this module is used to save or load a filter to of from a file. The files must have the extension “.mil” and their format is proprietary and text-based.
- *Export to Spice Module*: this module is used to export the filter into a spice-recognized file, thus offering the possibility to obtain a simulation.

- *Presentation Module*: this module is used to obtain several information about the filter such as sensitivity, residuum etc. Also a schematic view of the filter can be obtained using this module.
- *Derivative function module*: is used to compute the derivative functions for the filter.
- *Orthogonal function module*: is used to compute the orthogonal functions for the filter.
- *Orthonormal function module*: is used to compute the orthonormal functions for the filter.
- *Application Core*: this module is the bridge between all previous modules. It contains the application GUI and links to the functions implemented by all the previous modules.

Each of the previous modules is implemented as a set of C++ classes. All modules exchange data with the application core in both ways. One piece of data that is very important for the entire application is the transfer function represented as two polynomials. Each polynomial is saved within the application as a template class. The template for this class was needed to represent the polynomial coefficients. Usually those are real numbers, but the class allows the usage of complex values too.

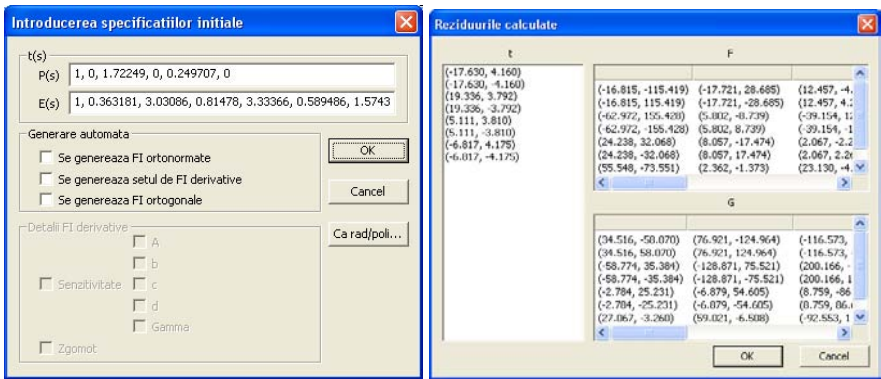


Fig. 13.3. Coefficients' Specification and Computed Residuum

Working with the application is a very easy task. After starting it the user can either choose to create a new filter or to open an existing one. The creation of a filter can be performed in two ways: either by entering the coefficients for the $P(s)$ and $E(s)$ polynomials (see Figure 13.3), or by specifying their degrees and complex zeroes. After the required data is entered, the user may decide which type of functions he/she want to generate. The application will generate the functions and the result will be displayed in a list form. The user can view information about IFs sets or about the

transfer function $t(s)$, such as the Residuum (see Figure 13.3), the user-friendly representation of the transfer function and its derivative. The elements of topological set $\{\mathbf{A}, \mathbf{b}, \mathbf{c}, \mathbf{d}\}$ and their computed sensitivity are also given. Another feature is the application's ability to generate a Spice-compatible file, based on which a simulation of the filter can be performed. In this way the user can further test the filter before deciding to implement in using real hardware.

13.4 Conclusion

The paper presented a C++ code program for the synthesis of continuous-time analog filters. The synthesis method is based on the selection of an appropriate set of intermediate transfer functions. This way the filter sensitivity, noise and dynamic range properties are easily determined from these intermediate functions. Therefore, it is possible to optimize the sensitivity, noise and dynamic range by the appropriate choice of intermediate functions. The synthesis program is easy to use and the user can select the optimum realization in rapport with one of the following performances: $\mathbf{A}$, $\mathbf{b}$, $\mathbf{c}$ and $\mathbf{d}$ sensitivities, noise and dynamic range. The solution helps the user to define better filters without using any expensive hardware components. He/she can generate several implementations for a filter and then, after examining the results using a simulation environment, can choose the best solution for his/she's needs. The user-friendly interface is very intuitive and the help integration offers further information about the usage of SSAF.

Acknowledgement. This work was supported by the CNCSIS Research Project no. 29C/08.05.2007, theme no. 5, CNCSIS code no.602.

References

1. Doicaru E (1998) The optimum automatic synthesis at active analogue continuous-time high-order filters. In: 6th International Symposium on Automatic Control and Computer Science, Volume II Computer Science, Iasi, Romania, pp 182-189
2. Press WH, Teukolski SA, Vetterling WT, Flannerry BP (1992) Numerical recipes in C: The art of scientific computing. Cambridge University Press, Cambridge
3. Snelgrove WM, Sedra AS (1980) A novel synthesis method for state-space active networks. In: Proc. Mid-west Symp. Circuits and Systems, Toledo

4. Snelgrove WM, Sedra AS (1986) Synthesis and analysis of state-space active filters using intermediate transfer functions. IEEE Transaction on Circuits and Systems CAS-33:287-300
5. Şabac IG (1981) Special Mathematics (in roumanian). Editura Didactica si Pedagogica, Bucuresti

Part 15

Theoretical approaches in information science

Information science has for a long time been drawing on the knowledge produced in psychology and related fields. This is reasonable, for the central issue in information science concerns individual users navigating information spaces such as libraries, databases, and the Internet. Because information science is about the definition and location of information, information seeking is the fundamental problem in information science, while other problems, such as document representation, are subordinate. In this part we present some theoretical approaches in the information science. The papers cover the complexity of the auxetic systems, new verification of reactive requirement for Lyee method, a tree derivation procedure for multivalent and paraconsistent inference, some results about feedback stabilization of matrix, Maxwell geometric dynamics, fractional right ideals, concurrent password recovery of archived files, solutions for an integral-differential equation, development of genetic algorithms and C-method for optimizing a scattering by rough surface.

Chapter 1

On the complexity of the auxetic systems

Ligia Munteanu,¹ Dan Dumitriu,¹ Ștefania Donescu,² Veturia Chiroiu¹

¹ Continuum Mechanics Department, Institute of Solid Mechanics,
Çtin Mille 15, Bucharest 010141

² Mathematics Department, Technical University of Civil Engineering,
Bd. Lacul Tei nr.122-124, Bucharest 020396

Abstract. Two major levels of complexity are discussed in a way of understanding the structure and processes that define an auxetic system. The auxeticity and structural complexity is interpreted in the light of Cosserat elasticity which admits degrees of freedom not present in classical elasticity, i.e. the rotation of points in the material, and a couple per unit area or the couple stress. The Young's modulus evaluation for a laminated periodic system made up of alternating aluminum and an auxetic material is an example of computing complexity.

Keywords. Auxetic material, Cosserat elasticity, Young's modulus, Negative Poisson's ratio, Homogenisation technique

1.1 Introduction

Materials with a negative Poisson ratio are auxetic materials. The term auxetic comes from the Greek word *auxetos*, meaning that *which may increase*. Instead of getting thinner like an elongated elastic band, the auxetic material grows fatter, expanding laterally when stretched. All the major classes of materials (polymers, composites, metals, ceramics honeycomb structures, reticulated metal foams, re-entrant structures, the skin covering a cow's teats, certain rocks and minerals, living bone tissue) can

exist in the auxetic form. Love [1] presents an example of cubic single crystal pyrite as having a Poisson's ratio of -0.14, and he suggests the effect may result from a twinned crystal. An *auxetic system* is composed from different materials with different properties with a new mechanical architecture based on tailored properties [2-5]. The idea is to transform a non-auxetic material into auxetic forms as foams or cellular materials, or to employ new architecture techniques for auxetic materials. The classical mechanics fail in describing the behavior of the auxetic material, because the chiral effects imply a fifth rank modulus tensor, which changes under an inversion. The chirality effects imply the first major level of complexity, i.e. *the structural complexity*. This structural complexity is described by Cosserat elasticity [6]. Phenomena associated with micropolar elasticity are likely to be of larger magnitude, and therefore of greater interest in materials such as auxetic systems with larger scale structural features [7-9]. The second level of complexity refers to *the computing complexity*. The estimation of the macroscopic Young modulus for a laminated periodic structure made up of alternating aluminum and an auxetic material layers, by using the homogenization technique, is an example of the computing complexity.

1.2 Structural complexity

Consider a chiral Cosserat medium, in a Cartesian coordinates system (x, y, z) . The equations of motion for the case without body forces and body couples are [10-12]

$$\sigma_{kl,k} - \rho \ddot{u}_l = 0, \quad m_{rk,r} + \varepsilon_{klr} \sigma_{lr} - \rho j \ddot{\phi}_k = 0. \quad (1.1)$$

Here σ_{kl} is the stress tensor, m_{kl} is the couple stress tensor, u is the displacement vector, ϕ_k is the microrotation vector which in Cosserat elasticity is kinematically distinct from the macrorotation vector $r_k = 1/2 \varepsilon_{klm} u_{m,l}$, and ε_{klm} is the permutation symbol. Note that ϕ_k refers to the rotation of points themselves, while r_k refers to the rotation associated with the movement of nearby points. In (1.1) ρ is the mass density and j the microinertia. The constitutive equations are

$$\begin{aligned} \sigma_{kl} = & \lambda e_{rr} \delta_{kl} + (2\mu + \kappa) e_{kl} + \kappa \varepsilon_{klm} (r_m - \phi_m) + \\ & + C_1 \phi_{r,r} \delta_{kl} + C_2 \phi_{k,l} + C_3 \phi_{l,k}, \end{aligned} \quad (1.2)$$

$$m_{kl} = \alpha \varphi_{r,r} \delta_{kl} + \beta \varphi_{k,l} + \gamma \varphi_{l,k} + C_1 e_{rr} \delta_{kl} + (C_2 + C_3) e_{kl} + (C_3 - C_2) \varepsilon_{klm} (r_m - \varphi_m), \quad (1.3)$$

where $e_{kl} = 1/2(u_{k,l} + u_{l,k})$ is the macrostrain vector, λ and μ are Lamé elastic constants, κ is the Cosserat rotation modulus, α, β, γ , the Cosserat rotation gradient moduli, and $C_i, i=1,2,3$, are the chiral elastic constants associated with noncentrosymmetry. For $C_i = 0$ the equations of isotropic micropolar elasticity are recovered. For $\alpha = \beta = \gamma = \kappa = 0$, (1.2) is reduced to the constitutive equations of classical isotropic linear elasticity theory [13-15]. Consider the case of the laminated plates made up of a periodic layering of sheets normal to the direction x of wave propagation, each of elastic material with constant properties. For simplicity, the particular 2D case in which all quantities depend only on x and z is considered. Let $F = \{\sigma_{kl}, m_{kl}, u_k, \varphi_k, k, l=1,2,3\}$ be a set composed of the asymmetric tensors $\sigma_{kl}, m_{kl}, k, l=1,2,3$, and the vectors u_k, φ_k . We call F an *elastodynamic state* on the bounded medium, if it satisfies (1.1)-(1.3) and appropriate initial conditions. There is one-by-one transformation, which transforms the elastodynamic state F into another elastodynamic state $\hat{F} = \{\hat{\sigma}_{kl}, \hat{m}_{kl}, \hat{u}_k, \hat{\varphi}_k, k, l=1,2,3\}$, composed by the symmetric tensors $\hat{\sigma}_{kl}, \hat{m}_{kl}, k, l=1,2,3$, and by the vectors $\hat{u}_k, \hat{\varphi}_k$, satisfying (1.1)-(1.3). The state $\hat{F}$ can be decomposed in $\hat{F} = \hat{F}_1 + \hat{F}_2$, where

$$\hat{F}_1 = \{\hat{\sigma}_{11}, \hat{\sigma}_{13}, \hat{\sigma}_{33}, \hat{m}_{22}, \hat{u}_1, \hat{u}_3, \hat{\varphi}_2\}, \quad \hat{F}_2 = \{\hat{\sigma}_{22}, \hat{m}_{11}, \hat{m}_{13}, \hat{m}_{33}, \hat{u}_2, \hat{\varphi}_1, \hat{\varphi}_3\}.$$

After a proper combination, the following equations in $\hat{u} = (\hat{u}_1, \hat{u}_2, \hat{u}_3)$ and $\hat{\varphi} = (\hat{\varphi}_1, \hat{\varphi}_2, \hat{\varphi}_3)$ are found

$$(\lambda + 2\mu + \kappa) \nabla \nabla \hat{u} - (\mu + \kappa) K_0^2 \nabla \times \nabla \times \hat{u} + \kappa(1 - K_0^2) \nabla \times \hat{\varphi} = \rho \ddot{\hat{u}}, \quad (1.4)$$

$$(\alpha + \beta + \gamma) \nabla \nabla \hat{\varphi} - \gamma K_0^2 \nabla \times \nabla \times \hat{\varphi} + \kappa(1 - K_0^2) \nabla \times \hat{u} - 2\kappa(1 - K_0^2) \hat{\varphi} = \rho j \ddot{\hat{\varphi}}, \quad (1.5)$$

with the coupling coefficient K_0 defined as

$$K_0^2 = 1 + \frac{(C_1 + C_2 + C_3)^2}{(\lambda + 2\mu + \kappa)(\alpha + \beta + \gamma)}. \quad (1.6)$$

One can see that (1.4) and (1.5) are decoupled into two sets of equations in $\hat{F}_1$ and respectively $\hat{F}_2$. We will concentrate only on the set of equations corresponding to $\hat{F}_1$, the other set being solved in a similar way.

1.3 Computing complexity

To solve the equations (1.4)-(1.6) with proper initial conditions, let us apply the Laplace and Fourier transforms, as follows

$$\begin{aligned}\tilde{v}_1'' &= \frac{b_s}{a^2} \tilde{v}_1 + \frac{i\xi(1-a^2)}{a^2} \tilde{v}_3' + \frac{s_4^*}{a^2} \tilde{\phi}_2', \quad \tilde{v}_3'' = c_s \tilde{v}_3 + i\xi s_4^* \tilde{\phi}_2 + i\xi(1-a^2) \tilde{v}_1', \\ b_s &= \xi^2 + \frac{p^2}{s_1 + s_2}, \quad c_s = a^2 \xi^2 + \frac{p^2}{s_1 + s_2}, \\ d_s &= \xi^2 + \frac{2c_1^2 \kappa(1-K_0^2)\mu}{\omega^2 \gamma K_0^2} + \frac{p^2}{s_4}.\end{aligned}$$

The eigenvalue problem is obtained by taking the solutions of the form

$$W(\xi, z, p) = X(\xi, p) \exp(qz), \quad (1.7)$$

where $W(\xi, z, p) = \{\tilde{v}_1, \tilde{v}_3, \tilde{\phi}_2\}$. The characteristic equation becomes

$$q^3 - \lambda_1 q^2 + \lambda_2 q + \lambda_3 = 0. \quad (1.8)$$

The roots of (1.8) are q_i , $i = 1, 2, 3$, with real parts positive. The eigenvector $X(\xi, p)$ is

$$X_{i1}(\xi, p) = \begin{vmatrix} a_i q_i \\ b_i \\ -\xi \end{vmatrix}, \quad X_{i2}(\xi, p) = \begin{vmatrix} a_i q_i^2 \\ b_i q_i \\ -\xi q_i \end{vmatrix}, \quad i = 1, 2, 3.$$

Consequently, the solutions (1.7) can be written as

$$W(\xi, z, p) = \sum_{i=1}^3 B_i X_i(\xi, p) \exp(q_i(\xi, p)z),$$

where B_i , $i = 1, 2, 3$, are arbitrary constants.

1.4 Case study and conclusions

Let us consider a laminated 2D composite plate which occupies the region $x \in [0, L]$, $z \in [-c, c]$, and made up by alternating the N aluminum and auxetic material layers, normal to the direction x of wave propagation (Figure 1.1). The layers are parallel, planar and periodic, the displacements being continuous across them. The length of each layer is l . The interfaces between layers are located at nl , $n = 1, 2, \dots, N$, each joint having two faces identified by $+$ and $-$. The coordinates are chosen so that the waves lie in the (x, z) plane. The plate is assumed to support in plane strains and waves running in the x -direction. The Bécus homogenization technique is applied via multiple scale expansion [16].

We are interested in knowing the influence of the Cosserat rotation modulus κ , the Cosserat rotation gradient moduli α, β, γ , and the chiral elastic constants C_i , $i = 1, 2, 3$, on the effective Young's modulus value of the laminated plate. The material constants for this laminated composite are periodic functions of x , i.e. $\tilde{C}(x + P) = \tilde{C}(x)$, where the period P equals to the length of the basic cell for the composite $P = 2l$, with $2l$ the period represented by the length of the basic cell for the composite.

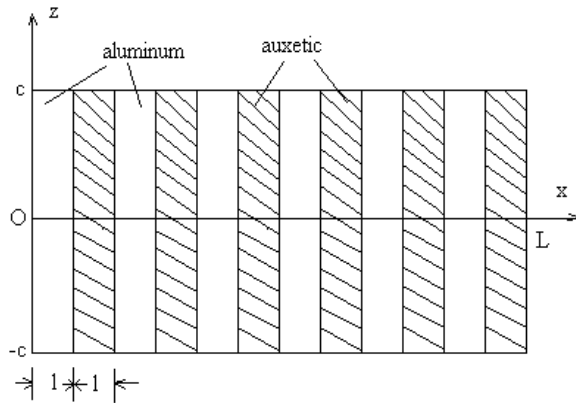


Fig. 1.1. The composite plate

The Bécus homogenization via multiple scale expansion yields

$$E = F(\tilde{C}') + E', \quad E' = \gamma + \frac{1}{2} \tilde{p}^2, \quad \tilde{p}^2 = \frac{2\kappa_{aux}}{(K_0'^2 - 1)}, \quad K_0'^2 = 1 + \zeta,$$

$$\gamma = \frac{(2\mu' + \kappa')(3\lambda' + 2\mu' + \kappa_{aux})}{(2\lambda' + 2\mu' + \kappa_{aux})},$$

$$\zeta = \frac{(C_{1aux} + C_{2aux} + C_{3aux})^2}{(\lambda' + 2\mu' + \kappa_{aux})(\alpha_{aux} + \beta_{aux} + \gamma_{aux})}.$$

Here $\tilde{C}_{al}$ are the aluminum constants and $\tilde{C}_{aux}$ the auxetic constants. The function $F(\tilde{C}')$ is numerically determined only. In the simulation, the Young's moduli of aluminum and respectively of the auxetic material are 109 GPa, and 1.55 GPa. We observe that the Young's modulus is increasing with respect to the volume fraction θ from 2 GPa to about 140 GPa, having a maximum value for $\theta = 0.75$. For θ above this value, the Young's modulus is decreasing with respect to θ from 140 GPa to about 90 GPa.

Acknowledgement. The authors acknowledge the financial support of the National University Research Council (NURC-CNCSIS) Romania, Grant nr. 55/2007 and Postdoctoral CEEX Grant nr. 1531/2006.

References

1. Love AEH (1926) A treatise on the mathematical theory of elasticity. 4th ed., Dover, New York
2. Lakes RS (1991) Experimental micro mechanics methods for conventional and negative Poisson's ratio cellular solids as Cosserat continua. J. Engineering Materials and Technology 113:148–155
3. Lakes RS (1987) Foam structures with a negative Poisson's ratio. Science 235:1038–1040
4. Lakes RS (1986) Experimental microelasticity of two porous solids. Int. J. Solids, Structures 22:55–63
5. Chiroiu V (2004) Identification and inverse problems related to material properties and behaviour. In: Chiroiu V, Sireteanu T (eds) Topics in Applied Mechanics. Ed. Academiei Bucharest, 1:83–126
6. Cosserat E and F (1909) Theorie des Corps Deformables. Hermann et Fils, Paris
7. Hlavacek M (1975) A continuum theory for fibre reinforced composites. Int. J. Solids and Structures 11:199–211
8. Hlavacek M (1975) On the effective moduli of elastic composite materials. Int. J. Solids and Structures 12:655–670
9. Berglund K (1982) Structural models of micropolar media. Mechanics of Micropolar Media, World Scientific, Singapore

10. Eringen AC (1968) Theory of micropolar elasticity. In: Liebowitz R (ed) Fracture. Academic Press, 2:621–729
11. Mindlin RD (1964) Microstructure in linear elasticity. Arch. Rat. Mech. Anal. 16:51–78
12. Mindlin RD (1965) Stress functions for a Cosserat continuum, Int J. Solids Structures 1:265–271
13. Gauthier RD (1982) Experimental investigations on micropolar media. Mechanics of Micropolar Media, World scientific, pp 395–463
14. Teodorescu PP, Munteanu L, Chiroiu V (2005) On the wave propagation in chiral media. New Trends in Continuum Mechanics, Ed. Theta Foundation, Bucharest, pp 303–310
15. Teodorescu PP, Badea T, Munteanu L, Onisoru J (2005) On the wave propagation in composite materials with a negative stiffness phase. New Trends in Continuum Mechanics, Ed. Theta Foundation, Bucharest, pp 295–302
16. Bécus GA (1979) Homogenization and random evolutions: Applications to the mechanics of composite materials. Quarterly of Applied Mathematics XXXVII(3):209–217

Chapter 2

New verification of reactive requirement for Lyee method

Osamu Arai, Hamido Fujita

Faculty of Software and Information Science, Iwate Prefectural University
Morioka, Iwate, JAPAN
arai@fujita.soft.iwate-pu.ac.jp, issam@soft.iwate-pu.ac.jp
<http://www.fujita.soft.iwate-pu.ac.jp/>

Abstract. Software development in general lacks simplicity and richness in expressing appropriately the requirement, as well related supportive tools. Despite the advances in this field, the solutions have still not overcome and the proposed way of thinking is far from resolving the problems on software development and maintenance. Recently, a new promising methodology, called Lyee, has been proposed. It aims to generate programs automatically from user requirement. Program structure in terms of Lyee was formalized. However, when requirements are reactive type, the validation mechanism has not proposed yet to show correctness or appropriateness of the running requirement. For this purpose, the properties that requirement and operation should have are described as a proposition in terms of temporal logic and are verified by such logic. New verification of reactive requirement for Lyee Method puts together as a design rule.

Keywords. Lyee, temporal logic, verification, reactive program, user requirement

2.1 Introduction

If requirement is well defined and appropriately expressed, software methodologies can realize them correctly for system built up. In order that the program runs correctly and ends, we should be able to clarify that requirement and operation should have a certain property for evaluation purposes.

The Lyee program generation principle [1] is applicable to Lyee requirement specification. This means that the program is either providing the user with the correct computation results, or indication to the user that the specification is incorrect. In Lyee methodology, requirement specification is described using Word and Process Root Diagram which is automatically expanded to Routing Vector (a kind of Word). The correctness of Lyee in the fixed-point setting [2], extension of this for parallel and distributed computing [3] is already discussed. Lyee requirement and program structure are formalized [4-6]. However, we think that almost all of Lyee style program behaves as reactive system program. In the case with multi screen and database management system or external equipments, the behavior of system is depend on the external employment such as human or external equipment reaction. Therefore, it is necessary that we apply another style of verification for such reactive requirement. As an example, Lyee requirements representing “The Light Control Case Study” [7] are discussed. In the case of Process Control System such as energy saving system of buildings, the users’ intension (requirement) is to save energy. Such intension (requirement) cannot verify without using process model.

In this paper, we consider the whole system including not only the structure of its program but also its external requirement and operation of human to behave as *reactive system*. We formalize whole system by process model, which consist of *processes*, *states* and *shared variables* as a bulletin board type communication tool. For this purpose, a language called L is defined by shared variables declaration and process definition embedded in its syntax. The semantics is represented by the *labeled transition system* extracted from the program. By constructing the *Temporal Frame* (TF) that is the semantics in terms of temporal logic extracted from the calculation sequence of that system, the linkage of semantics of temporal logic and that of L, is presented. Based on this provision, the property, (i.e., users’ intension) which the requirement specification should have, is described as a proposition in temporal logic. Such proposition is verified by the theorem of temporal logic, and its derivation rule.

The remainder of this paper is organized as follows. In Section 2.2, we introduce Lyee Methodology. In Section 2.3, we define the program of reactive system in terms of language L. We discuss the semantics of reac-

tive system in terms of language L and the semantics of temporal logic for validation purposes. In Section 2.4, we verify those properties in terms of Reactive program. In Section 2.5, we discuss classification of condition and fairness, and transition of state and transition of SF in PRD in Lyee method. As summary of discussion, the design rule is shown. Finally, Section 2.6 provides concluding remarks on this work, and related future work.

2.2 Lyee methodology and requirement

In the Lyee methodology, requirement specification is described using Word and Process Rout Diagram which is routing every SFs. SF (scenario function) is a unit in which inputs, process computations and outputs are reached in fixed-point representation. If SF is an object for screen, then the output is displayed waiting for input. With pushing the enter button after a value of variable input, it is restarted and processing is performed. Using transition of screen or database management system (DBMS), two or more SF are required. One SF is assigned to every single button of screen and to every single table of database management system. Accordingly, Process Route Diagram (PRD) is constructed. PRD expresses transition of two or more screens, the correlation of a screen and DBMS, etc. as transition between SFs. A routing vector is defined for this purpose. Transition of two or more SFs is defined as PRD. For more details on Lyee refer to [1].

In the case of screen and keyboard communication, Lyee requirement is almost have reactive property in itself or whole system including environment such as human operation have such property.

2.3 Reactive systems

In recent years, not only Lyee requirement, the computing system has developed into online system style computation. Such systems are called reactive system. Functional modeling cannot capture reactive system feature. Therefore, it is necessary to investigate all states during calculation. Since the sequence of these states continues indeterministically, a program does not have deterministic output value at the time of program end, so the formulization by input/output relation is not simple. A program continues reacting over input from outside.

To confirm whether the reactive system runs correctly as intended, even if such system itself does not have concurrent property, the whole system, which consists of system, and its exteriors have concurrent property.

2.3.1 Composition of reactive system

A reactive system consists of the following.

- Process

The subject that calculates by running a program is called a process. Reactive system consists of two or more calculation subjects called process, and each process runs in parallel, communicating with other processes.

- State

The global running status of a process that is named as *state* is used to grasp internal state of the system. The state includes not only allocation of value to variable but also working situation of process.

- Inter-process communication

Communication is done by shared variables or channels [4] that is representing message exchange among processes. We use a bulletin board type communication that uses the shared variable as communication means.

2.3.2 Reactive system on Lyee specification

We define a program L for reactive system and give its syntax and semantics for formal representation. Program has declaration of shared variables and definition of processes. Process definition has two attributes such as state and action. In declaration of a shared variable, this program declares to use the variable shared between each process. The other part of this program defines the action of a process. A process changes its state, by updating shared variables. An action of a certain process is defined as in the following: when a process is in state α , if the value of a shared variable fulfills Conditions B, performing substitution A of an assigned value to a shared variable then such process transfers itself to state β . The below formula shows such representation:

$$\alpha \xrightarrow[t]{B \Rightarrow A} \beta$$

where t is the name of action. Here after, program expressed using this representation, we call process model. The conditional expression B about

the value of a shared variable is called the execution conditions of t . The substitution of the value to a shared variable is called the command A of t . In Lyee Method structure, Scenario Function SF which has pallets W04-W02-W03 [1] in Figure 2.1 can be expressed by process model in Figure 2.2. Pallet W02 has two processes; one is computer (C), the other is external event such as human operation (H). If processes have more than two outputs, to describe which output should be selected, Process Rout Diagram is needed. Generally, output is selected by human operation or executed result of W03. The left part (a) of Figure 2.2 can be summarized to right part (b). This is because the variable domain for routing is limited inside of SF; W04-W02-W03. The requirement of reactive type system could be described as a transition of SF which is shown in (b).

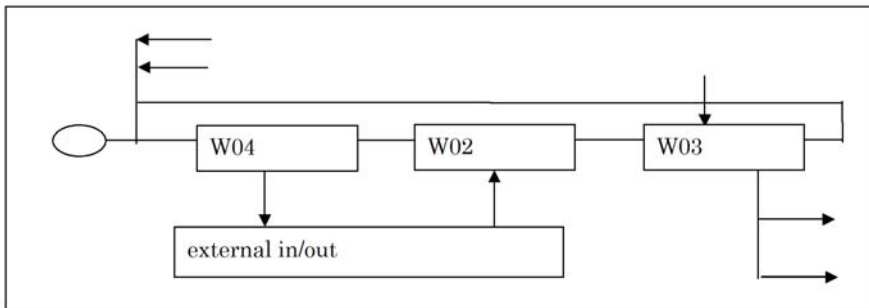


Fig. 2.1. Lyee Program

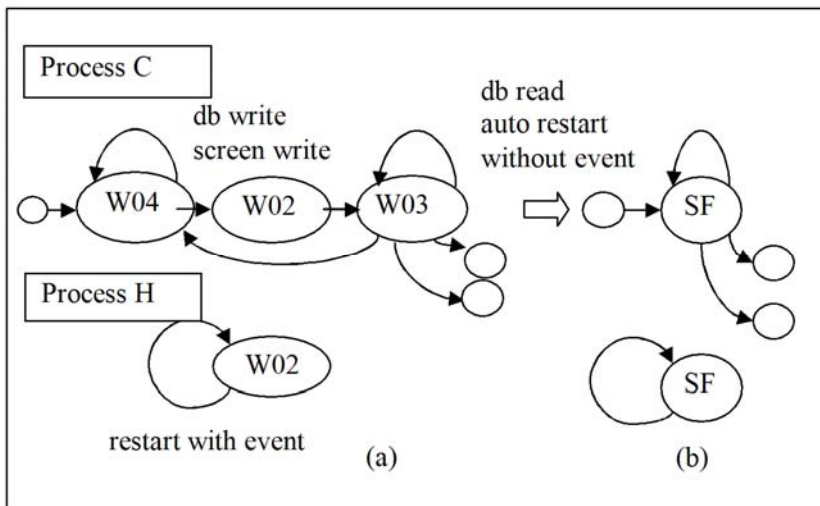


Fig. 2.2. Inner of SF

A sample program of reactive system type is shown as Figure 2.3 in full description, where two variables *start* and *finish* (includes initial value) are defined by $\{start:=undefined, finish:=false\}$. Two processes named H (Human) and C (Computer) react with each other using shared variables *start* and *finish*. At state *SF*, when shared variable *start* is false, process H transfer its state from *a* to *itself* changing *start* variable.

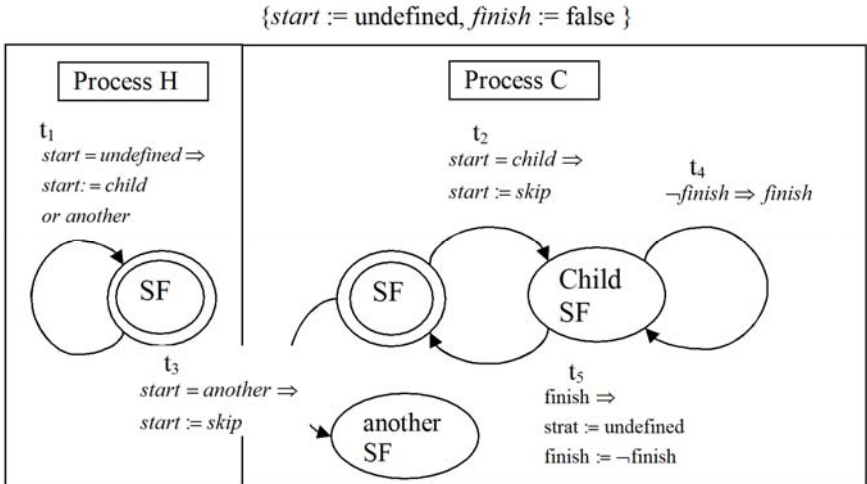


Fig. 2.3. Reactive System Program

Then Process C can transfer its state from *SF* to *Child SF* without changing any variables. At state *Child SF*, Process C can transfer its state from *Child SF* to *itself* changing variable *finish* from false to true when after memory reading is complete. In the case to transfer another SF, variable *start* have value *another*.

2.4 Verification Lyee requirement by temporal logic

To treat this representation more formally, we introduce a formal language L and temporal logic [8-10]. In the definition of program C in language L, its syntax and semantics is defined using Labeled Transition System (LTS). We introduce temporal logic to verify requirement described as a proposition of temporal logic. Time modality connector $\Box, \Diamond, \bigcirc$ means

$P := p$	(The notation showing a proposition)
$ (\neg P) (P \rightarrow Q)$	(not P) (if P then Q)
$ (P \wedge Q) (P \vee Q)$	(P and Q) (P or Q)
$ (\Box P)$	(P is true always)
$ (\Diamond P)$	(P is true some time)
$ (\bigcirc P)$	(P is true at the next time)

At definition of its semantics, we introduce temporal frame as a structure.

Generally, the fairness of processes execution should be maintained to some extent by scheduling. In Process route diagram, at the connecting SF, this fairness of execution is important. If the execution of routing is not fair, user's intension such as termination of program and saving energy result of process execution cannot achieved. For the verification of requirement, fairness is introduced [10].

Based on this provision, the intension that specification should have is described as a proposition of temporal logic, and is verified by the axiom of temporal logic, and related reduction rules. Lyee requirement specification which represents Light Control Case Study [7] is defined as an example; its intension is to save energy. The property of Lyee requirement (intension) specification is described as proposition of temporal logic.

2.4.1 Verification by temporal logic

Verification that a C program in terms of L-program satisfies a property P that is described by expression in temporal logical, means that P is universally valid in C. In this case, we argue about the property "*to be universally valid in the temporal frame made from the calculation sequence of LTS extracted from C*"; although it is not universally valid in any other temporal frame. For this reason, the following theorems are provided.

Theorem 1

Suppose that $P_1, \dots, P_n \models_{IM} P$ is true in a set IM of temporal frame and logical expressions $P_1, \dots, P_n$ and P . Then, $Q_1, \dots, Q_m \models_{IM} Q$ is true if Q can be extracted from $Q_1, \dots, Q_m$ by adding $P_1, \dots, P_n \vdash P$ as reduction rule.

IM is the set of a temporal frame. Moreover, the following general theorems are used for our use.

Theorem 2

Let P, Q be logical formula. $P \rightarrow \bigcirc Q$ is universally valid formula on C , when the statement [If P is realized before the execution of τ and then τ is performed, and Q will be true after execution of τ .] is realized to arbitrary transition τ in terms of LTS that is extracted from Program C .

Generally, liveness property is to prove the statement [a property Q will be satisfied at a certain time]; therefore the rules below are used.

If P is satisfied then P or Q will be satisfied at a certain time. If P continues to be satisfied then τ can be done at a certain time. If P is satisfied before execution of τ then Q is satisfied after execution. Therefore, it will be warranted that if P is satisfied then Q will be satisfied at a certain time. What formulized above as reduction rule of liveness shown below. This used to verify the property of specification that is written as a proposition in temporal logic.

Theorem 3 (rule of liveness)

The following are satisfied when the set $\{\tau\}$ which consists of transition τ of the LTS extracted from Program C are a weak fairness set or a strong fairness set on C .

$$\begin{aligned} &P \rightarrow \bigcirc (P \vee Q), \\ &P \rightarrow (Q \vee \text{enabled}_\tau), \\ &P \rightarrow \bigcirc (\text{executed}_\tau \rightarrow Q) \\ &\models_{\{M_{C,\sigma} \mid \sigma \in \llbracket C \rrbracket_{J,F}\}} P \rightarrow \Diamond Q \end{aligned}$$

$\models_{\{M_{C,\sigma} \mid \sigma \in \llbracket C \rrbracket_{J,F}\}}$ means that to the temporal frame $M_{C,\sigma}$, $P \rightarrow \Diamond Q$ is universally valid in terms of $C_{J,F}$ under strong and weak fairness where σ is a calculation sequence belonging to $\llbracket C \rrbracket_{J,F}$.

Theorem 4 (rule of chain)

The following is satisfied on arbitrary natural number i .

$$\begin{aligned}
& P_0 \rightarrow \Diamond Q \\
& P_1 \rightarrow \Diamond (P_0 \vee Q) \\
& \vdots \\
& P_i \rightarrow \Diamond (P_0 \vee P_1 \vee \dots \vee P_{i-1} \vee Q) \\
& \models (P_0 \vee P_1 \vee \dots \vee P_i) \rightarrow \Diamond Q
\end{aligned}$$

These two Theorems 3 and 4 are proven by deduction. According to the Theorem 1, we can use the formula, which used $\vdash$ instead of $\models$.

2.4.2 Example

Our Example is a part of program specification which represents the Light Control Case Study [7]. This case study is saving energy and security control of a building. Here we use energy saving part of this case study. That is «If nobody is in the room, light is turned off in fixed time». The Program is shown in Figure 2.4. Related processes and its states are shown in Table 2.1.

Shared variables shared by processes which control state transition and common variables that are used plural states commonly are defined. Declaration of shared variables and common variables are shown in Table 2.2. The state which means somebody deemed in the room made by events of different equipments is introduced. Process Definition is shown in Table 2.3. Since the variable which controls transition is expressed as a shared variable, then this variable is specified in command A. The shared variable is cleared by transition before turning into its initial state. s, t, z, x = 0, 0, 0, 0. States (SFs) c have two outgoing transitions (routes). In c, transition t6 and t8 are selectable. In this case, fare selection is needed. In order to save energy, program selects transition from c to a in a certain time.

2.4.3 Proof of example

The property which specification should have is described, as a proposition in terms of temporal logic, and is verified by the axiom of temporal logic and its derivation rule. The requirement that represents Light control case study is defined as an example. It is to verify that, *"its L-program surely save energy"*.

In this verification, we clarify the conditions of fairness, which requirement and operation should have. Liveness property of *"save energy"* is influenced by fairness property. Verification of Liveness is the proof of logi-

cal formula deploying fairness sets (defined in section 2.2). If people go out of the room, a timer will surely be finished after a while and the light will be put out. That is, fairness is required between the frequency f incomings and outgoings of people, and interval setting of a timer.

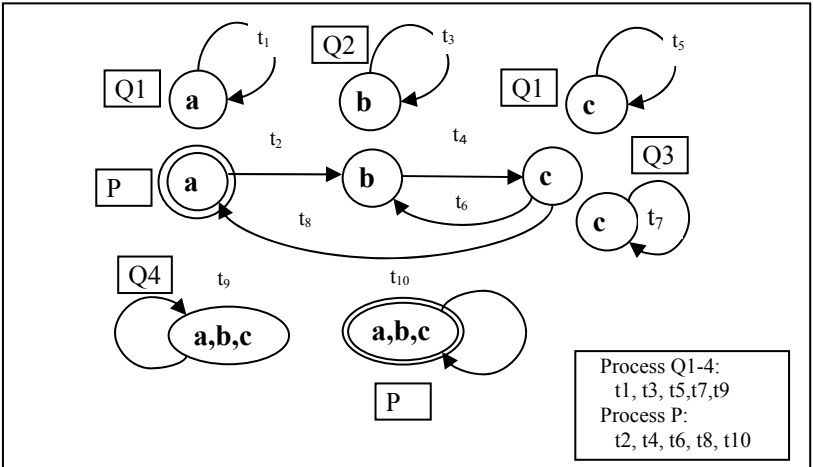


Fig. 2.4. Example of program Light Control Case

Table 2.1. Process and state of example

name	description
Process Q1	human detector on
Process Q2	human detector off
Process Q3	deemed-timer time up
Process Q4	light sensor scanning timer time up (repeat)
Process P	computer whose role is governor
State a	nobody in one room
State b	somebody in the room
State c	deemed-timer running
State a,b,c	deemed somebody in the room

Table 2.2. Declaration of shared and common variables

Declaration of shared and common variables	Description
shared variable	0: nobody in
$s = \{0,1\}$	1: somebody in
shared variable	0: deemed-timer stopped
$t = \{0,1\}$	1: running
shared variable	0: light sensor scan off
$x = \{0,1\}$	1: scan on

Table 2.3. Process definition

Process	τ	α	β	B	A
Q1	t_1	a	a	s, t = 0, 0	s, t, z = 1, 0, 0: some one detected
P	t_2	a	b	s, t = 1, 0	s, t, z = 1, 0, 1: deemed
Q4	t_9	a,b,c	a,b,c	x = 0	x = 1: light-sensor-scan on
P	t_{10}	a,b,c	a,b,c	x = 1	x = 0: light-sensor-scan off control room light on z

As the proposition of temporal logic, $[save\ energy]$ is expressed as: at $P.c \rightarrow \Diamond at P.a$. At this time, Processes P and Q are located c, and uses shared variables t (initial value of which is $\{t := 0\}$). For this reason, here after, $[at P.c \rightarrow \Diamond at P.a]$ is verified.

Using the state c of process P and the value of variable t , $at P.c$ is classified as in the following two cases:

$$at P.c = ((at P.c \wedge t = 0) \quad (2.1)$$

$$\wedge (at P.c \wedge t = 1) \quad (2.2)$$

Since $t = \{0, 1\}$, this classification is comprehensively inclusive and exclusively not overlapped internally in this case. Since the value of t is 0 in the case of (2.1), Process P can approach into a immediately. On the other hand, (2.2) is far from reaching the end In order to return to a , at timer is over, input 0 for t is required in Process Q.

(Apply Chain rule)

The name P_0, P_1 is given to each term according to the order near the goal ($at P.a$) for (2.1) to (2.2).

$$P_0 \equiv at P.c \wedge t = 0$$

$$P_1 \equiv at P.c \wedge t = 1$$

The following reduction will be obtained when the chain rule of the Theorem 4 is applied to such logical formula.

$$P_0 \rightarrow \Diamond at P.a,$$

$$P_1 \rightarrow \Diamond (P_0 \vee at P.a),$$

$$\vdash (P_0 \vee P_1) \rightarrow \Diamond at P.a$$

Liveness property is verifiable when the premise of rule from (2.1) to (2.2) is derived because $(P_0 \vee P_1)$ is $at P.c$.

(Apply Live rule)

Logical expressions P_0, P_1 named above are derived using the temporal logic shown below.

1) Derivation of (2.1)

Since P_0 is $at\ P.c \wedge t = 0$, when this is true, $\langle P, t_g \rangle$ is performed at a certain time and the state of P should be set to a . That is, fair execution is required for t_g as mentioned above. Therefore, in applying the live rule of Theorem 3 to $P_0 \rightarrow \Diamond at\ P.a$ and $\langle P, t_g \rangle$, the following derivation can be obtained:

$$P_0 \rightarrow o(P_0 \vee at\ P.a), \quad (2.3a)$$

$$P_0 \rightarrow (at\ P.a \vee enabled_{\langle P, t_g \rangle}), \quad (2.3b)$$

$$P_0 \rightarrow o(executed_{\langle P, t_g \rangle} \rightarrow at\ P.a) \quad (2.3c)$$

$$\models_{\{M_{C,\sigma} \mid \sigma \in [[C]]_{T,F}\}} P_0 \rightarrow at\ P.a$$

The logical expressions 2.3a, 2.3b, 2.3c are the premise of this derivation and verified as below.

(Verification of (2.3a))

In P_0 , since states of process P and Q are both $c = 0$ and $t = 0$, possible transition is restricted to $\langle P, t_g \rangle$; (case 1) and the idling transition τ_I ; (case 2).

Case 1: When $\langle P, t_g \rangle$ is performed, since variable $at\ P.c$ is true, the formula (a) is verified using Theorem 2.

Case 2: When τ_I is performed, since P_0 is true, the formula (2.3a) is true by Theorem 2.

Therefore, formula (2.3a) is verified. (2.3b), (2.3c) are verified same way.

Therefore, it is verified that formula (2.1), (i.e., $P_0 \rightarrow \Diamond at\ P.a$) turns into a universally valid formula of L-program. (2.1) is verified same way.

Since it is verified that (2.1) and (2.2) are universally valid formulas, the liveness property $at\ P.c \rightarrow \Diamond at\ P.a$ which is our objective, is verified.

□

Consequently, when satisfying the following requirements, the property that energy saving is performed correctly is verified under the fair execution.

1. Using the combination of shared variables, a special classification is necessary to carry out such execution. In this classification, condition must be comprehensively inclusive and exclusively defined. This means that the conditions of transition of SF is defined only once all over domain of a vector.
2. It is demanded that fair execution (i.e., Timer time up should surely be occurred some time) need to be done.

2.5 Discussions

From the state transition diagram (STR), extract SF pattern of STR according to event. Unify asynchronously simultaneous events (SF pattern of STR) to one SF. This is the SF division rule. One Parent SF is a group of events that may happen there. A screen (SF) has some buttons, i.e. events, in human interface with screen and keyboard, machine (sensor, actuator) interface event. If these events happen asynchronously, it needs to be received from one parent SF. That is, a group of events in which one of these events happens and the other event does not happen, compose one parent SF. Event transfers process from one state to another state (event driven). Therefore, more than one state exists in one parent SF.

2.5.1 Shared variable and common variable

In Process Control Application, there are many events such that communication that include internet, timer, human interface with screen and keyboard, machine (sensor, actuator) interface event. If these events happen asynchronously, it needs to be received from one parent SF. That is, a group of events in which one of these events happens and the other event does not happen, compose one parent SF. Event transfers process from one state to another state (i.e., event driven). Therefore, more than one state exists in one SF. The variable in one SF that is used commonly by plural states is called common variable.

2.5.2 Classification of condition and Fairness

In classification of transition condition, using the combination of shared variables needs fair classify. All conditions must be comprehensively inclusive and exclusively defined. This means that the conditions of transi-

tion of SF is defined only once all over domain of variable. Fair execution should be done. In the example, timer should surely time-up some time.

2.5.3 Expression as a design rule

The program in terms of Lyee structure is correctly, performed only after a definition and operation of requirement of Lyee are executed correctly. Calculation conditions of words and the conditions of transition of SF are the reference of requirement correctness.

PRD of Lyee specification is expressed in L-program by considering SF as a state using the process model. Since a state is an atomic action, SF cannot be partitioned into more than one atomic action. If SFs are unified and enlarged, an excessive calculation will be carried out and its execution speed may become slow. It is important to extract the boundary of SF appropriately. For this reason, it is effective to apply the process model proposed here for safety checking.

In order to apply the above discussion result from now on, it puts together as a design rule.

1. Describe a state transition diagram based on L-language.
2. Verify Property which design intension (requirement) wants.
3. Repeat 1) and 2) until fix the state transition.
4. Divide system to program (PRD).
5. Divide program (PRD) to SF based on the SF division rule above.
6. Draw PRD (transition between SF) based on state transition diagram.
7. Realize state as a set of common variable in SF by Lyee method such as memory table, action vector, and word.

Terms from 1 to 3 are about process model, term from 4 to 7 is about Lyee Methodology. According to this Design rule, we have developed full of the Light Control Case Study correctly and successfully by Lyee Method.

2.6 Conclusion

To verify the reactive property of requirement that multi screen, database management and process control system have, we should apply to reactive style of verification.

As an example, the requirement that represents Othello game is defined to verify that this program calculates correctly and surely stops anyway.

Consequently, we clarify the conditions of fairness in which the execution conditions of requirement and operation should have.

In classification presented at the proof of the example, conditions must be comprehensively inclusive and exclusively defined. Moreover, operation should be executed fairly. These requirements are common in PRD of Lyee methodology.

Especially, in Process control example, the Light Control Case Study, the trace of logic is difficult, and so maintainability is poor, the concept of state of process model is effective as an approach on visualizing and catching the program structure. When using a structured program such as Lyee or PROLOG, etc. for reactive application with many state transitions, this model is effective.

In Lyee Methodology, requirement specification is not verified. To use this methodology for actual system development, system design intension is desired to be verified beforehand. Using the process model, we can verify other reactive property of intension (requirement) such as safety, partial correctness, avoidance of dead lock, mutual exclusion of critical resource, total correctness, in the future.

References

1. Negoro F (2000) Principle of Lyee software. In: Proceedings of 2000 International Conference on Information Society in the 21st Century (IS2000), Aizu, Japan, pp 441-446
2. Gorlatch S (2003) The Lyee programming model: Analysis correctness in a fixed point setting. In: Fujita H, Johannesson P (eds) *New Trends in Software Methodologies, Tools and Techniques*. IOS Press, pp 214-224
3. Gorlatch S (2004) Declarative programming with Lyee for distributed systems. In: Fujita H, Gruhn V (eds) *New Trends in Software Methodologies, Tools and Techniques*. IOS Press, pp 129-137
4. Fujita H, Mejri M, Ktari B (2004) A process algebra to formalize the Lyee methodology. *Knowledge-Based Systems* 17(7-8):263-282, ISSN 0950-7051
5. Arai Osamu, Fujita Hamid (2003) Mathematical structure model for Word-Based Program. *Knowledge-Based Systems* 16(7-8): 399-411
6. Arai O, Fujita H (2006) Verification of the Lyee requirement. In: Fujita Hamid, Mejri Mohamed (eds) *New Trends in Software Methodologies, Tools and Techniques*. IOS Press, (Proceedings of the 5th SoMeT_06, Quebec), pp 340-361
7. Queins S et al. (2000) Requirement engineering: The light control case study. *Journal of Universal Computer Science* 6(7)
8. Manna Z, Pnueli A (1991) *The temporal logic of reactive and concurrent systems*. Springer-Verlag

9. Manna Z, Pnueli A (1995) The temporal verification of reactive system; Safety. Springer-Verlag
10. Manna Z, Pnueli A (1996) The temporal verification of reactive system; Progress. Draft

Chapter 3

A tree derivation procedure for multivalent and paraconsistent inference

David Anderson

School of Computing, University of Portsmouth, Portsmouth PO1 2EG,
Ph. (+44) 2392 84 6668, david.anderson@port.ac.uk

Abstract. The use of trees as a derivation procedure, which is a refinement of Beth's semantic tableaux method and Gentzen's sequent calculus, is an elegant and well established technique. Despite widespread acceptance and application, little effort has been made to extend the use of tree derivation procedures to multivalent alternatives to classical logic and none at all, so far as I am aware, to paraconsistent systems. The purpose of this paper is to outline briefly a single tree derivation procedure which was designed to work with a particular multivalued paraconsistent system: Epsilon₄₄₂ [1], but will, with slight modification, work generally.

Keywords. Automated reasoning, Logic, Semantic tableaux, Paraconsistency, Sequent Calculus

3.1 Valid arguments

In classical systems, a valid argument may be distinguished from an invalid argument by virtue of the fact that the former has no counter examples. Hence, if an exhaustive search fails to produce a counter example we may be certain that an argument is valid and otherwise invalid. The first method for carrying out such a search which most of us encounter is the 'truth table' but a much more elegant approach is the construction of logical trees.

In classical propositional logic the following familiar rules will suffice:

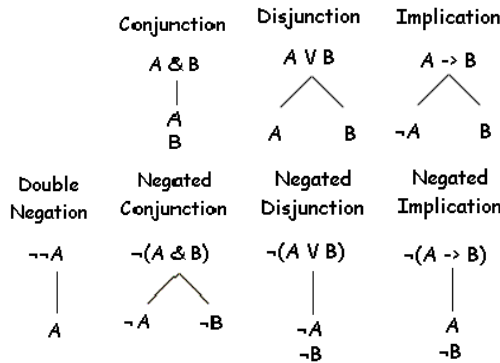


Fig. 3.1. Tree derivation rules for classical logic

3.2 Tree construction

A basic process for constructing a tree derivation is as follows:

- Step 1: Arrange the premisses in a column one above the other.
- Step 2: To the column formed in Step 1, add the NEGATION of the conclusion.
- Step 3: Using the decomposition rules presented above simplify each of the expressions (or nodes) as far as possible.
- Step 4: When a node has been simplified, it is no longer of interest and may be crossed out, ticked or otherwise identified as no longer forming an active part of the structure.

A trivial example follows. Given $A \rightarrow B$, $A \mid\text{-} B$, the application of steps 1-4 yield the following structure:

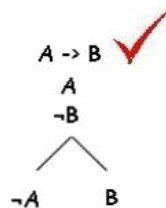


Fig. 3.2. An example of inference using tree derivation with classical logic

Each path from top to bottom through the tree we have created represents a possible valuation demonstrating the satisfiability of the set of

statements under consideration.

The test for argument validity is carried out in:

Step 5: We put a cross at the bottom of each branch which contains any given statement AND its negation. A branch so constituted is called closed and any tree all of branches are closed is itself said to be closed.

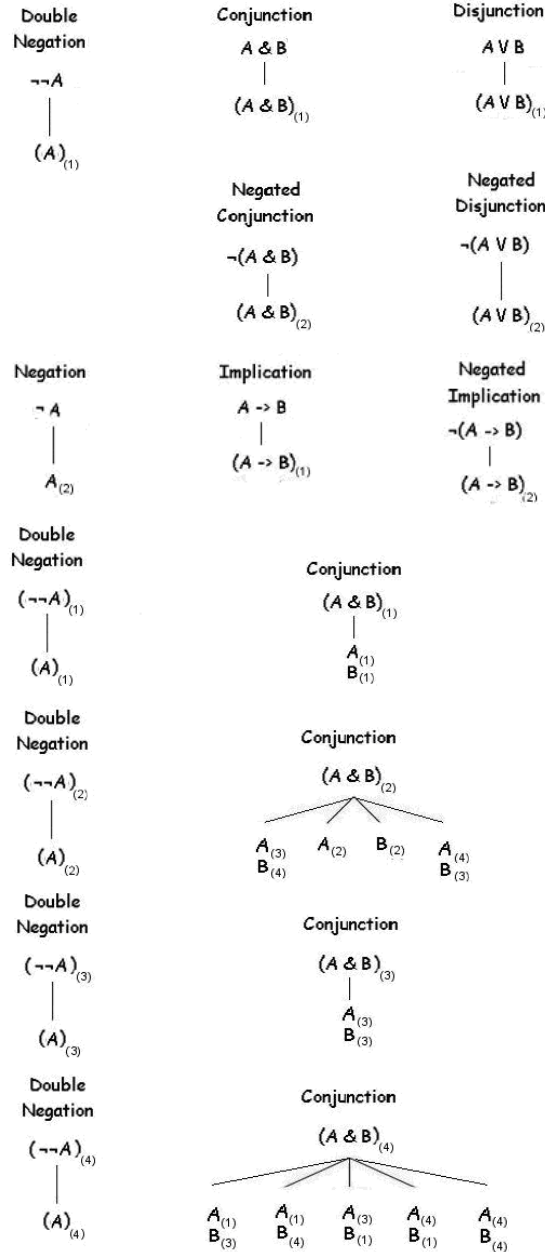
A valid inference will always give rise to a closed tree and invalid inferences produce trees with one or more open branches. This is because the decomposition rules are simply ways of displaying the conditions under which a given complex expression may be true and, since we have included the negation of the conclusion as part of the structure, each branch represents a possible counter-example to the original inference. A closed tree is one where all the possible counter-examples have been eliminated and hence demonstrates validity.

3.3 Multivalent tree construction

If this general approach is to be extended to cover multivalued systems there are a number of considerations which need to be addressed. First, we need to exercise care over the use of the negation sign, ' $\neg$ '. In the system just described $\neg$ stands as a marker for falsity (an its absence indicates truth) as well as a logical operator. In a multivalued system matters are apt to be more complicated and it will prove more effective to avoid any possible ambiguity and use a new piece of notation to indicate the valuation being asserted at any given point on the tree. To this end, I will employ subscripted parentheses containing the valuation. Thus to indicate that A is true we would employ the following: $A_{(true)}$, and for A is false, we would use $A_{(false)}$. In what follows I will be outlining a system designed to handle a four-valued system (more details of which are given in Appendix A) whose possible valuations are 1,2,3 and 4 hence $A_{(1)}$, $A_{(2)}$, $A_{(3)}$ and $A_{(4)}$ will be used to make clear which of the available valuations A takes at any given point.

Another consideration to bear in mind is that multivalued systems give rise to greater complication concerning how to interpret valuations. In particular, decisions about which valuations are to be designated, anti-designated or non-designated mean that what counts as a counter-example is not as clearcut as in the bivalent system. One possible approach to this would be to construct a new tree for each of the sorts of counter example but this leads to a more cumbersome system and the preferred approach is

to rely on a single tree at the expense of slightly more convoluted tree derivation process. The rules given below were developed for Epsilon 442, in which {1,3} are designated values, {2} is anti-designated and {4} is non-designated. The full matrices for Epsilon 442 are given in Appendix.



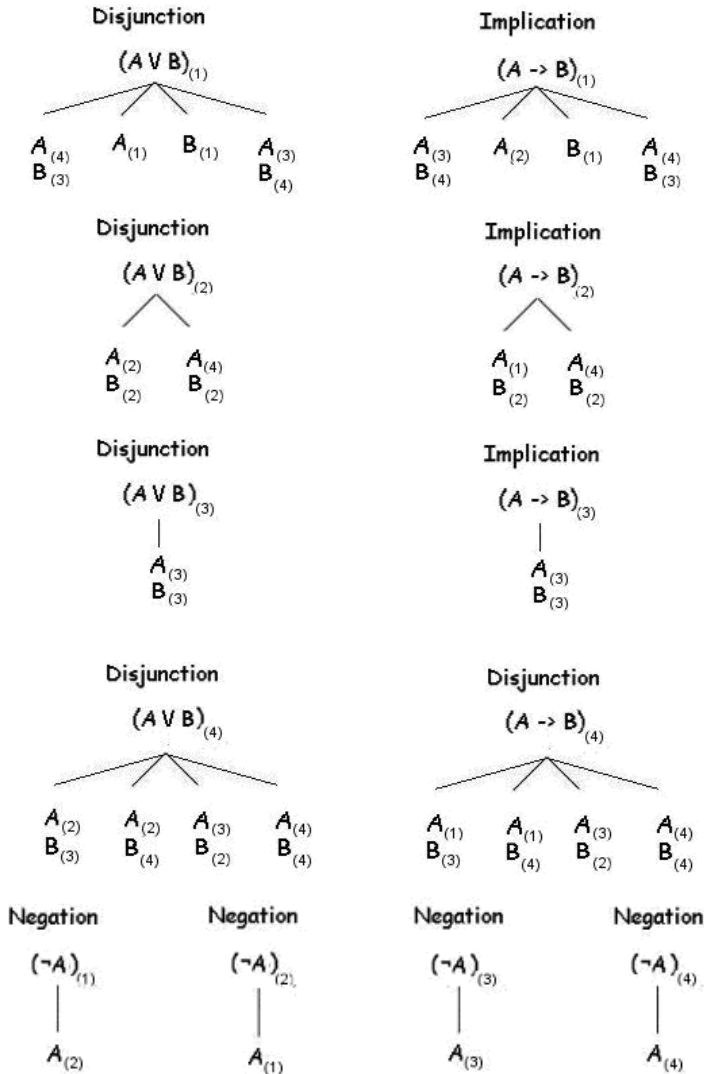


Fig. 3.3. Tree derivation rules for a 4-valued logic Epsilon 442

How do we arrive at exactly these rules and how would we develop rules for a different system?

Rather than painstakingly going through each structure in turn I will take a single example in order to demonstrate the principle. Take $(A \& B)_{(4)}$. To devise the appropriate decomposition structure we need to consult the Epsilon 442 matrix for conjunction:

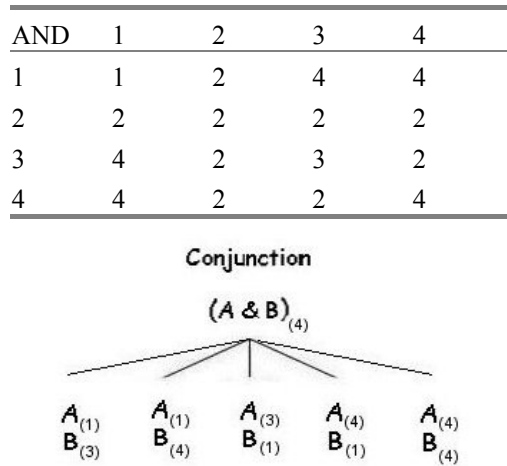


Fig. 3.4. Devising an appropriate tree structure from an Epsilon 442 matrix

We can readily see that $(A \& B)$ is assigned the value 4 in just five cases:

- i. the first operand has a value of $_{(1)}$ and the second has a value of $_{(3)}$.
- ii. the first operand has a value of $_{(1)}$ and the second has a value of $_{(4)}$.
- iii. the first operand has a value of $_{(3)}$ and the second has a value of $_{(1)}$.
- iv. the first operand has a value of $_{(4)}$ and the second has a value of $_{(1)}$.
- v. the first operand has a value of $_{(4)}$ and the second has a value of $_{(4)}$.

And these five cases are captured precisely by the decomposition rule where each of the alternatives is represented by its own branch. Each of the other structures is arrived at in a similar fashion. Of course, different matrices would give rise to a different decomposition rule but the process remains straightforward.

3.3.1 Multiple tree derivation scheme

- Step 1: Conjoin all the premisses and mark the resulting conjunction as having the lowest numbered available designated valuation, e.g. $_{(1)}$.
- Step 2: Below the expression formed in Step 1, add an expression which is the original conclusion and mark it as having the lowest numbered available anti-designated valuation, e.g. $_{(2)}$.
- Step 3: Using the decomposition rules presented above simplify each of the expressions (or nodes) as far as possible.

- Step 4: When a node has been simplified, it is no longer of interest and may be crossed out, ticked or otherwise identified as no longer forming an active part of the structure.
- Step 5: Put a cross at the bottom of each branch which contains any given statement having two different valuations e.g., $A_{(3)}$ and $A_{(1)}$. A branch so constituted is called *closed* and any tree all of branches are closed is itself said to be closed

When steps 1-5 have been completed, if the tree is closed, the process should be repeated with the next unused designated valuation (in this case (3)) being employed in Step 1. This process should continue until there are no untested designated valuations. Then the whole cycle should be gone through again using the next unused anti-designated valuation (in this system there is only one anti-designated valuation) being employed in Step 2, and so on until all the anti-designated valuations have been tested.

A valid inference will always give rise to a closed tree for every test and invalid inferences produce one or more trees with one or more open branches.

3.3.2 Single tree derivation scheme:

- Step 1: Conjoin all the premisses and mark the resulting conjunction as having the lowest numbered available designated valuation, e.g. (1) . Repeat the process with each remaining unused available designated valuation, the resulting conjunctions should be disjoined with each other to form one complex disjunction.
- Step 2: Mark the original conclusion and as having the lowest numbered available anti-designated valuation, e.g. (2) . Repeat the process with each remaining unused available anti-designated valuation, the resulting expressions should be disjoined with each other to form one complex disjunction. Add the resultant expression below the expression formed in Step 1.
- Step 3: Using the decomposition rules presented above simplify each of the expressions (or nodes) as far as possible.
- Step 4: When a node has been simplified, it is no longer of interest and may be crossed out, ticked or otherwise identified as no longer forming an active part of the structure.
- Step 5: Put a cross at the bottom of each branch which contains any given statement having two different valuations e.g., $A_{(3)}$ and $A_{(1)}$. A branch so constituted is called *closed* and any tree all of branches are closed is itself said to be closed

A valid inference will always give rise to a closed tree and invalid inferences produce one or more open branches.

It is worth noting that the tree derivation scheme suggested here handles a multivalent system which is also paraconsistent and in this respect it is, to the best of my knowledge, unique.

References

1. Anderson CDP (2002) Developing a framework for investigating inconsistency handling in automated reasoning. In: 6th World Multi-Conference on Systems, Cybernetics and Informatics (SCI 2002), Orlando, Florida

Appendix

Matrices for Epsilon 442:

NOT					AND 1 2 3 4				
1	2				1	1	2	4	4
2	1				2	2	2	2	2
3	3				3	4	2	3	2
4	4				4	4	2	2	4
¬(¬P & ¬Q)					¬(P & ¬Q)				
OR 1	2	3	4		COND 1	2	3	4	
1	1	1	1	1	1	1	2	4	4
2	1	2	4	4	2	1	1	1	1
3	1	4	3	1	3	1	4	3	1
4	1	2	1	4	4	1	2	1	4

NB. In Epsilon 442, {1,3} are designated values, {2} is anti-designated and {4} is non-designated.

Chapter 4

On the feedback stabilization of matrix

JianDa.Han, Zhe.Jiang, YiWen.Zhao, YiYong.Nie

State Key Laboratory of Robotics, Shenyang Institute of Automation,
Academia Sinica. 110016 Shenyang, China

Abstract. In this paper, several strategies for feedback stabilization of matrix are presented via the numerical stabilization of polynomial and matrix.

Keywords. Feedback stabilization, numerical stabilization, determining coefficients, Hessenberg form, Frobenius-like form

4.1 Introduction

It is important to guarantee stability of a dynamic system. In order to design a stable control system, the numerical stabilization of system has become quite an interesting problem.

Every state equation of linear time-invariant system can be described as follows,

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (4.1)$$

For a system with p inputs, and n state variables, A and B are, respectively, $n \times n$ and $n \times p$ constant matrices. In state feedback, the input vector u is given by $u = r - Kx$, where K is a $p \times n$ gain matrix with real elements k_{ij} . This is the constant gain negative state feedback. We may suppose $r = 0$ since only stability of the state feedback system is investigated. Substituting $u = -Kx$ into the state equations of (4.1) yields

$$\dot{x} = (A - BK)x = \begin{pmatrix} a_{11} - v_{11} & a_{12} - v_{12} & \cdots & a_{1n} - v_{1n} \\ a_{21} - v_{21} & a_{22} - v_{22} & \cdots & a_{2n} - v_{2n} \\ & & \cdots & \\ a_{n1} - v_{n1} & a_{n2} - v_{n2} & \cdots & a_{nn} - v_{nn} \end{pmatrix} x, \quad (4.2)$$

where

$$v_{ij} = \sum_{s=1}^p b_{is} k_{sj}, \quad i, j = 1, 2, \dots, n. \quad (4.3)$$

The matrix $A - BK$ is referred to as stable if its each eigenvalue has negative real-part. The system (4.1) is asymptotically stable if and only if the matrix $A - BK$ is stable; see, e.g., [3,5,6]. The matrix $A - BK$ is referred to as critical if the largest real-part of eigenvalues is equal to zero. If the system (4.1) is critical, then the second approximation may be constructed such that (4.1) is stable or unstable, see, e.g., [5]. So that the critical matrix is rather considered unstable. A useful matrix for control system must have sufficient stability, namely, the largest real-part of eigenvalues must be less than or equal to an appropriate negative number.

Regarding each k_{sj} in (4.3) as a parameter, we can appropriately select $k_{sj} (s = 1, \dots, p, j = 1, \dots, n)$ such that the feedback system (4.2) is stable. This is referred to as *feedback stabilization*. Evidently, simultaneously selecting k_{sj} is too complicated for general A and B . Fortunately, in order to stabilize the matrix A by the state feedback matrix K , we need only to investigate some simple cases of B and K .

The stability of a given matrix is equivalent to the stability of its eigenpolynomial. In fact, using numerically stabilized elementary transformations we can first reduce a matrix to upper Hessenberg form, then reduce upper Hessenberg form to Frobenius-like form, and finally we can obtain the eigenpolynomial. In computation process, the reduction from initial matrix to Frobenius-like form may be numerically stable for initial perturbation and rounding errors, see [2,6,7].

In the same reason, the largest real-part of roots of the eigenpolynomial must be less than or equal to an appropriate negative number.

In this paper, so-called *numerical stabilization* of polynomial and matrix means to regulate the changeable coefficients of polynomial or the changeable elements of matrix such that the largest real-part of roots or eigenvalues

lues, ξ , is less than or equal to an appropriate negative number, the *stabilized-threshold* θ ,

$$\xi \leq \theta, \quad (4.4)$$

while original polynomial or original matrix is possibly unstable ($\xi > \theta$) or over stable ($\xi \square \theta$) in itself, see [1].

The stabilized-threshold θ is a key to numerical stabilization. If (4.4) holds for polynomial or matrix after stabilization, then the polynomial or matrix is properly stable, otherwise unstable or over stable. Clearly $\theta < -2^{-t}$, where t is the mantissa digits of computer, because $2^{-t} \approx 0$ in computer. Moreover, since all the stability criteria are based on the calculations of polynomial coefficients or matrix elements, θ must be dependent on the accumulation of relative rounding-errors in the computational process. Unfortunately, the accumulation of rounding-errors is a statistical processes. We can only estimate a error bounds too high, even if the algorithm is numerically stable.

The error analyses of the numerically stable algorithms, including the reduction from initial matrix to Frobenius-like and the quasi-Routh array, show that the error bounds $O(n2^{-t})$ relates to elements of $A - BK$. This is inexpedient. In order to avoid including elements of $A - BK$, we rather regard an enlarged error bounds as the stabilized-threshold θ in (4.4). That is,

Hypothesis 1 *For the numerically stable algorithms with $O(n^3)$ flops, including the reduction from initial matrix to Frobenius-like and the quasi-Routh array, the stabilized-threshold in (4.4)*

$$\theta = -n^3 2^{-t} \quad (4.5)$$

where n is dimension of the system (4.1), and t is the mantissa digits of computer.

If $t = 52$ and $n = 50, 100, 150$, then $\theta \approx -2.7756 \times 10^{-11}$, -2.2204×10^{-10} , -7.4940×10^{-10} , respectively. If $t = 32$ and $n = 30, 40, 50$, then $\theta \approx -6.2864 \times 10^{-6}$, -1.4901×10^{-5} , -2.9104×10^{-5} respectively. Practical computation shows that the stabilized-threshold given by (4.5) is appropriate for numerical stabilization of polynomial and matrix.

It is easy to see a simple strategy for feedback stabilization of (4.2).

Proposition 2 *The proper stabilization of the matrix (4.2) may be always realized by the origin-shifted transformations. Namely the matrix A in*

(4.1) may be always stabilized by the diagonal feedback matrix $K = g_d I$, where I denotes unit matrix and g_d is a real number (diagonal gain).

Proof: Let $P(\lambda) = \det(A - \lambda I)$ be the eigenpolynomial of the matrix A in (4.1), where $\det(\square)$ denotes determinant of matrix, and let l be the bounds of the eigenvalues of A . Using the origin-shifted transformation, we know that $P(\lambda + l) = \det(A - (\lambda + l)I) = \det((A - lI) - \lambda I)$ is stable and $P(\lambda - l) = \det((A + lI) - \lambda I)$ unstable.

Let $s_0 = 0$. If $P(\lambda + s_0)$ is unstable or critical, then let $s_1 = s_0 + l/2$, otherwise let $s_1 = s_0 - l/2$, then investigate the stability of $P(\lambda + s_1)$. If $P(\lambda + s_1)$ is unstable or critical, then let $s_2 = s_1 + l/2^2$, otherwise let $s_2 = s_1 - l/2^2$, then investigate the stability of $P(\lambda + s_2)$ If $P(\lambda + s_{r-1})$ is unstable or critical, then let $s_r = s_{r-1} + l/2^r$, otherwise let $s_r = s_{r-1} - l/2^r$. We may always make $P(\lambda + s_r)$ stable or critical, $P(\lambda + s_r - l/2^r)$ unstable, and $P(\lambda + s_r + l/2^r)$ stable. When r is big enough, $l/2^r \leq -\theta$, and the origin-shifted transformation may be terminated. Thus, the largest real-part of matrix eigenvalues, $\xi \approx s_r$, may be found by a serial origin-shifted transformations.

Let $B = I$, $K = g_d I$ and $g_d = s_r - \theta$ in (4.2). Clearly $P(\lambda + s_r - \theta) = \det((A - BK) - \lambda I)$ is stable, and $\xi \approx g_d + \theta$. The matrix $A - BK$ is properly stable. $\square$

With numerical determination of matrix stability, Proposition 2 gives a basic approach to feedback stabilization. And the length of numerically experimental interval decreases with experimental times in exponent. However, the diagonal feedback matrix $K = g_d I$ requires n inputs with the same gain parameter g_d . This is possibly inconvenient. We must look for more strategies for feedback stabilization.

4.2 Numerical stabilization of polynomial and matrix

In this paragraph, we list all the useful results of numerical stabilization of polynomial and matrix, see [1].

Let

$$F(\lambda) = -\det(A - \lambda I) = \lambda^n + a_1 \lambda^{n-1} + a_2 \lambda^{n-2} + \cdots + a_{n-1} \lambda + a_n \quad (4.6)$$

be the monic polynomial with n degree. The determining coefficients $\alpha_i = \alpha_{i-1} \alpha_{i-2} / (\alpha_i \alpha_{i+1})$, $i = 1, \dots, n-2$, $\alpha_0 = 1$. Using the determining coefficients, we can present several strategies for stabilization of the point-wise polynomial (4.6).

Proposition 3. *If some invariable coefficients of (4.6) are given such that a determining coefficient $\alpha_i \leq 0$ or $\alpha_i \geq 1$, $1 \leq i \leq n-2$, at least, then the polynomial (4.6) is impossible to be stabilized.*

Generally, if there are many invariable coefficients in (4.6), then, in Proposition 3, it easily happens that (4.6) is impossible to be stabilized.

Suppose now (4.6) may be stabilized, then $0 < \alpha_i < 1$, ($i = 1, 2, \dots, n-2$). In order to look for more efficient strategies for numerical stabilization of (4.6), we utilize the sufficient conditions which are nearing critical condition, the optimum criteria of polynomial stability.

Hypothesis 4. *The polynomial (4.6) is considered a properly stabilized polynomial if one of the following two equalities holds, where and $\alpha_i = \alpha_{i-1} \alpha_{i-2} / (\alpha_i \alpha_{i+1})$ and $0 < \alpha_i < 1$*

$$(i) \max \{\alpha_1, \alpha_2, \dots, \alpha_{n-2}\} = 0.46557;$$

$$(ii) \max \{\alpha_1 + \alpha_2, \alpha_2 + \alpha_3, \dots, \alpha_{n-2} + \alpha_{n-1}\} = 0.88988.$$

Via Kharitonov's theorem, see [4], the stability of monic interval polynomial

$$F(\lambda) = \lambda^n + a_1 \lambda^{n-1} + a_2 \lambda^{n-2} + \cdots + a_{n-1} \lambda + a_n, \underline{a}_i \leq a_i \leq \bar{a}_i, i = 1, 2, \dots, n, \quad (4.7)$$

is equivalent to the stability of the following four point-wise polynomials:

$$F_1(\lambda) = \lambda^n + \underline{a}_1 \lambda^{n-1} + \bar{a}_2 \lambda^{n-2} + \bar{a}_3 \lambda^{n-3} + \underline{a}_4 \lambda^{n-4} + \underline{a}_5 \lambda^{n-5} + \cdots, \quad (4.8)$$

$$F_2(\lambda) = \lambda^n + \bar{a}_1 \lambda^{n-1} + \bar{a}_2 \lambda^{n-2} + \underline{a}_3 \lambda^{n-3} + \underline{a}_4 \lambda^{n-4} + \bar{a}_5 \lambda^{n-5} + \cdots,$$

$$F_3(\lambda) = \lambda^n + \underline{a}_1 \lambda^{n-1} + \underline{a}_2 \lambda^{n-2} + \bar{a}_3 \lambda^{n-3} + \bar{a}_4 \lambda^{n-4} + \underline{a}_5 \lambda^{n-5} + \cdots,$$

$$F_4(\lambda) = \lambda^n + \bar{a}_1 \lambda^{n-1} + \bar{a}_2 \lambda^{n-2} + \underline{a}_3 \lambda^{n-3} + \underline{a}_4 \lambda^{n-4} + \bar{a}_5 \lambda^{n-5} + \cdots.$$

Each polynomial of (4.8) has $n-2$ determining coefficients α_i ($i = 1, \dots, n-2$), while $\underline{\alpha}_i = \underline{a}_{i-1} \underline{a}_{i+2} / (\bar{a}_i \bar{a}_{i+1})$, and $\bar{\alpha}_i = \bar{a}_{i-1} \bar{a}_{i+2} / (\underline{a}_i \underline{a}_{i+1})$ are the least one and the largest one respectively.

Proposition 5 *If some invariable intervals of (4.7) are given such that a determining coefficient of (4.8), $\underline{\alpha}_i \leq 0$ or $\bar{\alpha}_i \geq 1$, $1 \leq i \leq n-2$, at least, then the interval polynomial (4.7) is impossible to be stabilized.*

Suppose (4.7) may be stabilized, then $0 < \underline{\alpha}_i \leq \bar{\alpha}_i < 1$ ($i = 1, \dots, n-2$).

Proposition 6 *If each point-wise polynomial of (4.8) is stabilized by Hypothesis 4, then the interval polynomial (4.7) is stabilized numerically.*

If the changeable elements of the matrix K in (4.2) is arbitrarily distributed, then one of the most reliable strategy for numerical stabilization of $A - BK$ is to reduce from $A - BK$ to the Frobenius-like form and to apply stabilized strategies of the eigenpolynomial with a lot of numerical experiments. Here, the numerical stability of each numerical experiment is important, otherwise the coefficients of eigenpolynomial are possibly distorted by accumulation of rounding-errors. Fortunately, the numerically stable algorithm of reduction from $A - BK$ to Frobenius-like form has been found with stabilized elementary similarity transformations, see [6,7].

Since the stability of A is only dependent on n invariants, it is reasonable to assume that there are only n changeable elements in n^2 elements of BK at most. Generally, with elementary similarity transformations, these n elements may be put on the first row of matrix. Therefore we assume further that the n changeable elements are distributed on the first row of BK . The numerical stabilization of $A - BK$ becomes more efficient on this assumptions.

Proposition 7 *If the changeable elements of $A - BK$ in (4.2) are on the first row, then the numerical stabilization of the matrix may be realized by the stabilized strategies of polynomial, Proposition 3 and Hypothesis 4.*

4.3 Feedback stabilization of interval controllable canonical form

Suppose $A = A_c$, $p = 1$, $B = B_c = (0, \dots, 0, 1)^T$ and $K = K_c = (k_1, \dots, k_n)$ in (4.1) and (4.2), where the superscript T denotes transposition and A_c is an interval controllable canonical form given by

$$A_c = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ & & & \cdots & & \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ a_1 & a_2 & a_3 & \cdots & a_{n-1} & a_n \end{pmatrix} \quad \alpha_i \in [\underline{a}_i, \bar{a}_i], i = 1, 2, \dots, n. \quad (4.9)$$

The n -dimensional state equation of controllable canonical form with single-input is as follows,

$$\dot{x}(t) = A_c x(t) + B_c u(t). \quad (4.10)$$

After feedback $u = -K_c x$ of single-input, the state equations (4.10) becomes

$$\dot{x} = (A_c - B_c K_c) x = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & 1 \\ a_1 - k_1 & a_2 - k_2 & \cdots & a_n - k_n \end{pmatrix} x, \quad (4.11)$$

where $k_i, i = 1, 2, \dots, n$, are regarded as undetermined parameters for stabilization of $A_c - B_c K_c$. The matrix $A_c - B_c K_c$ is still a canonical form.

Clearly $(A_c - B_c K_c)^T$ is a Frobenius-like form. Its monic eigenpolynomial is

$$P_c(\lambda) = \lambda^n + (k_n - a_n)\lambda^{n-1} + \cdots + (k_2 - a_2)\lambda + (k_1 - a_1). \quad (4.12)$$

Therefore all the stabilized strategies of polynomial may be applied to the interval controllable canonical form $A_c - B_c K_c$ in (4.11).

Proposition 8 *For the interval controllable canonical form (4.9), we may set $k_{n-1} = k_{n-1}^c$ and $k_n = k_n^c$ and then select $k_1, \dots, k_{n-2}$ such that the interval controllable canonical form $A_c - B_c K_c$ in (4.11) is stable.*

Proof: Setting $k_n > a_n$ and $k_{n-1} > a_{n-1}$ in any point-wise polynomial of (4.12) and using Proposition 3 and Hypothesis 4(i), we can obtain the fol-

lowing selected intervals of other undetermined parameters, $n-2 \geq n+2-2i$, $n+1-2i \geq 2$,

$$0 < k_{n+2-2i} - a_{n+2-2i} \leq (k_n - a_n)(\alpha^{i-1}(k_{n-1} - a_{n-1}))^{i-1}, \quad (4.13)$$

$$0 < k_{n+1-2i} - a_{n+1-2i} \leq (\alpha^{i-1}(k_{n-1} - a_{n-1}))^i, \quad (4.14)$$

where $\alpha = \max\{\alpha_1, \alpha_2, \dots, \alpha_{n-2}\} \leq 0.46557$. Evidently, from (4.13) and (4.14), $k_j \geq \bar{a}_j - \theta$ ($j=1, 2, \dots, n$), where θ is the stabilized-threshold. Let $k_j = \bar{a}_j + c_j$ ($c_j \geq -\theta$, $j=1, 2, \dots, n$). Then $k_j - a_j = \bar{a}_j - a_j + c_j$ and $k_j - a_j \in [c_j, d_j + c_j]$, where $d_j = \bar{a}_j - \underline{a}_j$ denotes the length of j -th interval. The coefficient intervals of (4.12) are $[c_j, d_j + c_j]$, $j=1, 2, \dots, n$. Let $d = \max\{d_1, d_2, \dots, d_n\}$.

It is easy to see that, from (4.14), the maximal value of function $f(i) = (\alpha^{i-1}(k_{n-1} - a_{n-1}))^i$ occurs close to some i^* , $4 \leq 2i^* \leq n$. This requires $\alpha^{3-2i^*} < k_{n-1} - a_{n-1} < \alpha^{-2i^*-1}$, namely $\alpha^{3-2i^*} < c_{n-1} < \alpha^{-2i^*-1} - d_{n-1}$. We hope also that, from (4.13), the maximal value of function $g(i) = (k_n - a_n)(\alpha^{i-1}(k_{n-1} - a_{n-1}))^{i-1}$ occurs close to i^* . This requires $\alpha^{4-2i^*} < c_{n-1} < \alpha^{-2i^*} - d$. Therefore we get

$$\alpha^{3-2i^*} < c_{n-1} < \alpha^{-2i^*} - d_{n-1}. \quad (4.15)$$

Suppose $\alpha^{-2i^*} + d_{n-1} > \alpha^{3-2i^*}$. Then i^* may be equal to $2, 3, \dots$ for $d_{n-1} < 20.160198$. Let $m = \lfloor (n+1)/2 \rfloor$ be the biggest integer less than or equal to $(n+1)/2$, and

$$c_{n-1}^* = (d + c^o)^{1/i^*} / \alpha^{i^*-1}, i^* \leq m, \quad (4.16)$$

$$c_n^* = (d + c^e) / (\alpha^{i^*-1} c_{n-1}^*)^{i^*-1} = (d + c^e)(d + c^o)^{1-1/i^*}, \quad (4.17)$$

where $c^o = \max\{c_{n-1}, c_{n-3}, \dots\}$ and $c^e = \max\{c_n, c_{n-2}, \dots\}$ are the prearranged parameters which must satisfy, with (4.15),

$$\alpha^{-(i^*-2)} < (d + c^o)^{1/i^*} < \min\{\alpha^{i^*-1} c^o, \alpha^{-(i^*+1)} - \alpha^{i^*-1} d_{n-1}\}, \quad (4.18)$$

$$(2 \leq i^* \leq m),$$

and

$$d \leq ((d + c^o)^{1-1/i^*} - 1)c^e, \quad (m \geq 3). \quad (4.19)$$

The inequalities (4.18) and (4.19) yield, with $c^o + d = \alpha^{-i^*-u}$ and $\alpha^{-u} - \alpha^{-u/i^*} \geq \alpha^{i^*} d$,

$$\max\{\alpha^{-i^*(i^*-2)} - d, \alpha^{i^*-u/i^*} + \theta\} < c^o < \quad (4.20)$$

$$\alpha^{-i^*(i^*+1)} (1 - \alpha^{2i^*} d_{n-1})^{i^*} - d < \alpha^{-i^*(i^*+1)} - d,$$

and

$$c^e \geq d / ((d + c^o)^{1-1/i^*} - 1). \quad (4.21)$$

Besides c^o and c^e satisfy (4.20) and (4.21) respectively, they must satisfy, with (4.16) and (4.17),

$$\alpha^{(m-1)(m-i^*)} (d + c^o)^{(m-i^*)/i^*} (d + c^e) > d, \quad (4.22)$$

and

$$\alpha^{m(m-i^*)} (d + c^o)^{m/i^*} > d \quad (4.23)$$

for $n = 2m$, or

$$\alpha^{(m-1)(m-1-i^*)} (d + c^o)^{(m-1)/i^*} > d \quad (4.24)$$

for $n = 2m - 1$. It is easy to see $i^* > (m-1)/2$ from (4.20), (4.23) and (4.24). Let $i^* = \lceil m/2 \rceil$ be the least integer bigger than or equal to $m/2$. Then $\alpha^{-i^*(m-i^*)} < \alpha^{-i^*(i^*+1)}$.

Let

$$c^o = \alpha^{-i^*(m-i^*)} \max\{1, d^{i^*/(m-1)}\} - d - \theta. \quad (4.25)$$

Then (4.20), (4.23) and (4.24) hold for $m \geq 4$ and $d \leq 4.6$. Let

$$c^e = \max\{\alpha^{-(i^*-1)(m-i^*)}, \alpha^{-(i^*-1)(m-i^*)} d^{(i^*-1)/(m-1)}, 2d\} - d - \theta. \quad (4.26)$$

Then (4.21) and (4.22) hold for $m \geq 3$. Substituting (4.25) and (4.26) into (4.16) and (4.17) respectively, we can obtain the representations of c_{n-1}^* and c_n^* , which are only dependent on α, d and m . For instance, with $m \geq 5$,

$$c_{n-1}^* = \alpha^{-(m-1)} d^{1/(m-1)} - \theta, c_n^* = 1, \quad (4.27)$$

where $1 \leq d \leq 4.6$. When $d < 1$, the equality (4.27) is still right if d is regarded as 1.

Setting $k_{n-1} = k_{n-1}^c = \bar{a}_{n-1} + c_{n-1}^*$ and using the Kharitonov's theorem, we may select $c_{n-3}, c_{n-5}, \dots, c_{n-2m+3}, c_{n-2m+1}$, namely $k_{n-3}, k_{n-5}, \dots, k_{n-2m+3}, k_{n-2m+1}$, satisfying (4.14) because of (4.15). For instance, via (4.8), $c_{n-3} \leq (\alpha(d_{n-1} + c_{n-1}^*))^2$ and $d_{n-3} + c_{n-3} \leq \alpha(c_{n-1}^*)^2$, so that

$$c_{n-3} \approx \alpha(c_{n-1}^*)^2 - d_{n-3}.$$

Also via (4.8), $d_{n-5} + c_{n-5} \leq (\alpha^2(d_{n-1} + c_{n-1}^*))^3$ and $c_{n-5} \leq (\alpha^2 c_{n-1}^*)^3$, so that

$$c_{n-5} \approx \min\{(\alpha^2 c_{n-1}^*)^3, (\alpha^2(d_{n-1} + c_{n-1}^*))^3 - d_{n-5}\}.$$

Similarly, setting $k_{n-1} = k_n^c = \bar{a}_n + c_n^*$ and using the Kharitonov's theorem, we may select $c_{n-2}, c_{n-4}, \dots, c_{n-2m+4}, c_{n-2m+2}$, namely $k_{n-2}, k_{n-4}, \dots, k_{n-2m+4}, k_{n-2m+2}$, satisfying (4.13) because of (4.16). For instance, via (4.8), $c_{n-2} \leq (c_n^* + d_n)\alpha c_{n-1}^*$ and $d_{n-2} + c_{n-2} \leq c_n^* \alpha c_{n-1}^*$, so that

$$c_{n-2} \approx c_n^* \alpha c_{n-1}^* - d_{n-2}.$$

Also via (4.8), $d_{n-4} + c_{n-4} \leq (c_n^* + d_n)(\alpha^2 c_{n-1}^*)^2$ and $c_{n-4} \leq c_n^* (\alpha^2 c_{n-1}^*)^2$, so that

$$c_{n-4} \approx \min\{c_n^* (\alpha^2 c_{n-1}^*)^2, (c_n^* + d_n)(\alpha^2 c_{n-1}^*)^2 - d_{n-4}\}.$$

After $c_{n-3}, \dots, c_1$ are represented by (4.8), (4.14) and (4.13), we should test and verify the stability of (4.12) with Kharitonov's theorem and Hypothesis 4(i). If the stability cannot be guaranteed, then α should be reduced gradually. If $d_i = d, i = 1, \dots, n$, then we need only to test and verify

$$\begin{aligned} \max\{\alpha_1\} &= (c_{n-2} + d)/(c_{n-1}^* c_n^*) \leq 0.46557, \\ \max\{\alpha_2\} &= (c_n^* + d)(c_{n-3} + d)/(c_{n-1}^* c_{n-2}) \leq 0.46557, \\ \max\{\alpha_{n-3}\} &= (c_2 + d)(c_5 + d)/(c_3 c_4) \leq 0.46557, \end{aligned}$$

$$\max\{\alpha_{n-2}\} = (c_1 + d)(c_4 + d)/(c_2 c_3) \leq 0.46557$$

because of (4.14) and (4.13). $\square$

Example 1 Suppose $n = 10$ and

$$a_1 \in -9 \pm 0.8, a_2 \in -11 \pm 0.8, a_3 \in -20 \pm 0.8, a_4 \in -4 \pm 0.8,$$

$$a_5 \in -6 \pm 0.8, a_6 \in -10 \pm 0.8, a_7 \in -5 \pm 0.8, a_8 \in -7 \pm 0.8,$$

$$a_9 \in -1 \pm 0.8, a_{10} \in -2 \pm 0.8$$

in (4.9). Clearly $m = 5, d = 1.6, i^* = 3$ and $d_1 = \dots = d_{10} = d$.

In this example, via (4.27), $c_9^* = 1.6^{1/4} \alpha^{-4}$ and $c_{10}^* = 1$.

According to Proposition 8 we obtain

$$c_8 = \alpha c_9^* - 1.6 = 1.6^{1/4} (\alpha^{-3} - 1.6^{3/4}),$$

$$c_7 = (\alpha c_9^*)^2 - 1.6 = 1.6^{1/2} (\alpha^{-6} - 1.6^{1/2}), \quad c_5 = 1.6^{3/4} \alpha^{-6},$$

$$c_4 = 1.6^{3/4} \alpha^{-3} - 1.6, \quad c_3 = (\alpha^3 * c_9^*)^4 - 1.6 = 1.6(\alpha^{-4} - 1),$$

$$c_2 = (\alpha^4 * c_9^*)^4 = 1.6,$$

$$c_1 = (\alpha^4 * (c_9^* + 1.6))^5 - 1.6 = (1.6^{1/4} + 1.6 \alpha^4)^5 - 1.6.$$

The α should also satisfy

$$1.6^{1/4} \alpha^{-3} / (1.6^{1/4} \alpha^{-4}) = \alpha \leq 0.46557,$$

$$2.6 * 1.6^{1/4} \alpha^{-2} / (1.6^{1/4} \alpha^{-3} - 1.6) \leq 0.46557,$$

$$(1.6^{1/4} + 1.6 \alpha^4)^5 * 1.6^{3/4} \alpha^{-3} / 1.6 / (1.6 \alpha^{-4} - 1.6) \leq 0.46557,$$

$$3.2 * (1.6^{3/4} \alpha^{-6} + 1.6) / (1.6 \alpha^{-4} - 1.6) / (1.6^{3/4} \alpha^{-3} - 1.6) \leq 0.46557,$$

Namely

$$\alpha \leq 0.1790653846(1 - 1.6^{3/4} \alpha^3),$$

$$\alpha(1 + 1.6^{3/4} \alpha^4)^5 \leq 0.46557(1 - \alpha^4),$$

$$\alpha(1 + 1.6^{1/4} \alpha^6) \leq 0.232785(1 - \alpha^3 - 1.6^{1/4} \alpha^4 + 1.6^{1/4} \alpha^7).$$

Evidently $\alpha = 0.46$ cannot satisfy the above inequalities. Since $\alpha < 0.46$, $1.6^{3/4} \alpha^4 \square 1$, we get approximately

$$\alpha + 0.2547426292 \alpha^3 \leq 0.1790653846, \quad \alpha \leq 0.1776,$$

$$\alpha + 0.46557 \alpha^4 + 5 * 1.6^{3/4} \alpha^5 < 0.46557,$$

$$\alpha + 3.737604604 \alpha^4 < 0.47557, \quad \alpha \leq 0.38,$$

$$\alpha + 0.232785 \alpha^3 + 0.26180925 \alpha^4 \leq 0.232785, \quad \alpha \leq 0.2292.$$

Select $\alpha = 0.1776$. According to Proposition 6 we have thus $c_9^* = 1130.467678664$ and $c_1 \approx 0.2122628$, $c_2 \approx 1.6$,

$$\begin{aligned}c_3 &\approx 1606.6299, \quad c_4 \approx 252.3575, \quad c_5 \approx 45334.8536769, \\c_6 &\approx 1271.417385, \quad c_7 \approx 40307.4184, \quad c_8 \approx 199.1710597.\end{aligned}$$

Therefore

$$\begin{aligned}k_{10} - a_{10} &\in [1, 2.6], \\k_9 - a_9 &\in [1130.467678664, 1132.067678664], \\k_8 - a_8 &\in [199.1710597, 200.7710597], \\k_7 - a_7 &\in [40307.4184, 40309.0184], \\k_6 - a_6 &\in [1271.417385, 1273.017385], \\k_5 - a_5 &\in [45334.8536769, 45336.4536769], \\k_4 - a_4 &\in [252.3575, 253.9575], \\k_3 - a_3 &\in [1606.6299, 1608.2299], \\k_2 - a_2 &\in [1.6, 3.2], \\k_1 - a_1 &\in [0.2122628, 1.8122628]\end{aligned}\tag{4.28}$$

in the interval polynomial (4.12). It is easy to test and verify that the interval polynomial (4.12) with coefficients (4.28) is proper stable since

$$\max\{\alpha_1, \dots, \alpha_8\} \leq \max\{\alpha_2\} \approx 0.46557$$

The parameters of feedback stabilization in this example are as follows,

$$\begin{aligned}k_1 &= -7.9877372, \quad k_2 = -8.6, \quad k_3 = 1587.4299, \quad k_4 = 249.1575, \\k_5 &= 45329.6536769, \quad k_6 = 1262.217385, \quad k_7 = 40303.2184, \\k_8 &= 192.9710597, \quad k_9 = 1130.267678664, \quad k_{10} = -0.2.\end{aligned}$$

4.4 Single feedback stabilization

If $B = (b_1, b_2, \dots, b_n)^T$ and $K = (k_1, \dots, k_n)$ in n -dimensional state equation with single-input (4.10), then the state equation of single feedback becomes

$$\dot{x} = (A - BK)x = \begin{pmatrix} a_{11} - b_1 k_1 & a_{12} - b_1 k_2 & \cdots & a_{1n} - b_1 k_n \\ a_{21} - b_2 k_1 & a_{22} - b_2 k_2 & \cdots & a_{2n} - b_2 k_n \\ & & \cdots & \\ a_{n1} - b_n k_1 & a_{n2} - b_n k_2 & \cdots & a_{nn} - b_n k_n \end{pmatrix} x, \quad (4.29)$$

where $k_i (i = 1, \dots, n)$ are undetermined parameters. We can appropriately select $k_i (i = 1, \dots, n)$ such that the system (4.29) is stable. However, simultaneously selecting k_i is evidently too complicated for numerical experiments. For general A and B , we can select single k_i successively.

Let $K = (0, \dots, 0, k_s, 0, \dots, 0)$ in (4.29). Then (4.29) is simplified into

$$\dot{x} = (A - BK)x = \begin{pmatrix} a_{11} & \cdots & a_{1s} - b_1 k_s & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2s} - b_2 k_s & \cdots & a_{2n} \\ & & \cdots & & \\ a_{n1} & \cdots & a_{ns} - b_n k_s & \cdots & a_{nn} \end{pmatrix} x. \quad (4.30)$$

We want to appropriately select k_s such that the stability of system (4.30) is improved with all possible. Let E_{1s} be the matrix commuting 1- s row/column. Then $\bar{A}_{ks} = (E_{1s} A_{ks} E_{1s})^T$ has the same stability as A_{ks} . The changeable elements of $\bar{A}_{ks}$ have been put on the first row of $\bar{A}_{ks}$. With the algorithm given by [1, 6], $\bar{A}_{ks}$ is reduced to the Frobenius-like form.

$$F_l = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & p_0 \\ 2^{\beta_2} & 0 & 0 & \cdots & 0 & p_1 \\ & 2^{\beta_3} & 0 & \cdots & 0 & p_2 \\ & & \cdots & & & \\ & & & 2^{\beta_n} & p_{n-1} \end{pmatrix}, \quad (4.31)$$

where $p_{n-1} = p_{n-1}^0 + \gamma_{n-i} k_s, i = 1, \dots, n$. Let $r_{n-i} = \sqrt{(p_{n-i}^0)^2 + \gamma_{n-i}^2}$. Then $p_{n-i} = r_{n-i} (p_{n-i}^1 + \gamma_{n-i}^1 k_s), i = 1, \dots, n$.

Each determining coefficient of (4.31) has become the following ratio

$$\alpha_i = \frac{r_{n-i+1}r_{n-i-2}}{r_{n-i}r_{n-i-1}} \frac{\eta_{i0} + \eta_{i1}k_s + \eta_{i2}k_s^2}{\xi_{i0} + \xi_{i1}k_s + \xi_{i2}k_s^2}, \quad (r_n = 1, 1 \leq i \leq n-2).$$

Using Proposition 3 and Hypothesis 4(i), we obtain that

$$f_i(k_s) = p_{n-i}/r_{n-i} = p_{n-i}^1 + \gamma_{n-i}^1 k_s < 0, \quad i = 1, \dots, n, \quad (4.32)$$

where every $f_i(k_s)$, $1 \leq i \leq n$ is a normalized linear function, and that

$$g_i(k_s) = q_{i0} + q_{i1}k_s + q_{i2}k_s^2 \leq 0, \quad i = 1, \dots, n-2, \quad (4.33)$$

where $q_{ij} = (r_{n-i+1}r_{n-i-2})/(r_{n-i}r_{n-i-1})\eta_{ij} - 0.46557\xi_{ij}$, $j = 0, 1, 2$. The necessary condition (4.32) is easily satisfied if all $\gamma_{n-i}^1 \neq 0$. The sufficient condition (4.33) usually is $n-2$ conics which give the conclusion whether k_s can be selected from some real interval $[-l, l]$.

If there is an appropriate real number k_s^* such that both (4.32) and (4.33) are satisfied, and

$$y(k_s^*) = \max\{y(k_s) = \max\{g_i(k_s) \leq 0 \mid 1 \leq i \leq n-2, k_s \in [-l, l]\}\}, \quad (4.34)$$

then the system (4.30) may be properly stabilized by single feedback k_s^* . If there is no $k_s \in [-l, l]$ such that (4.32) and (4.33) are satisfied, then we can select the best k_s^* such that

$$z(k_s^*) = \min\{z(k_s) = \min\{f_j(k_s), g_i(k_s) \mid 1 \leq j \leq n, 1 \leq i \leq n-2, k_s \in [-l, l]\}\}, \quad (4.35)$$

therefore the stability of (4.30) is improved with all possible. The k_s^* is referred to as *gain of single feedback stabilization*.

For general matrices A and B , we first design the gain of single feedback stabilization k_1^* for $A_1 = A$. If $A_{k_1^*}^1$ is still unstable, then let $A^2 = A_{k_1^*}^1$. For A^2 we design the gain of single feedback stabilization k_2^* by (4.35) or (4.34), and so on. If $A_{k_{s-1}^*}^s$ is still unstable, then let $A^s = A_{k_{s-1}^*}^s$. For A^s we design the gain of single feedback stabilization k_s^* by (4.35) or (4.34). (If $s = n$ and k_s^* is given by (4.35), then go on the next cycle of

design, $s+1=1$.) We may expect that $A^{s+1} = A_{k_s^*}^s$ will be properly stabilized, or that the stability of A^{s+1} cannot be improved any more by single feedback BK . ($k_1^* = k_2^* = \dots = k_n^* = 0$.) Usually, if $k_1^* \approx k_2^* \approx \dots \approx k_n^* \approx 0$ in one cycle of design and A^{s+1} is still unstable, then A cannot be stabilized by single feedback BK .

It is possible that effect of the above single feedback stabilization is different with different order of single feedback stabilization. Generally speaking, the single column feedback (4.30) is hard to stabilize a matrix.

Example 2 Suppose a general matrix is given by

$$A_8 = \begin{pmatrix} 1 & 2 & 1 & 4 & 5 & 3 & 2 & 1 \\ 1 & 3 & 2 & 5 & 3 & 2 & 3 & 2 \\ 2 & 4 & 1 & 7 & 2 & 1 & 3 & 1 \\ 3 & 5 & 6 & 1 & 3 & 4 & 1 & 2 \\ 2 & 7 & 3 & 2 & 1 & 2 & 5 & 1 \\ 4 & 5 & 1 & 3 & 7 & 3 & 2 & 6 \\ 3 & 3 & 5 & 7 & 1 & 2 & 4 & 5 \\ 2 & 2 & 4 & 1 & 3 & 2 & 2 & 1 \end{pmatrix}. \quad (4.36)$$

and

$$B_8 = (2, 4, 5, 7, 8, 1, 3, 5)^T. \quad (4.37)$$

For A_8 and B_8 , gains of single feedback stabilization

$$k_1^* = 3.98, \quad k_2^* = 2.05, \quad k_3^* = 2.0, \quad k_4^* = -1.54, \\ k_5^* = -0.34, \quad k_6^* = -0.13, \quad k_7^* = -0.07, \quad k_8^* = 0.29$$

are selected by (4.35) in the first cycle of single feedback stabilization, while A_8^{s+1} is still unstable. We consider that the stability of A_8^{s+1} cannot be improved by single feedback $B_8 K^*$.

On the other hand, we know, via the Gerschgorin's disk theorem, that $G_8 = A_8 - 31I$ is possibly over stable, where I denotes unit matrix. The matrix G_8 can be made a properly stable matrix by single feedback stabilization.

For instance, with (4.32)-(4.34) we may design the gain of single feedback stabilization g_1^* for the over stable matrix $G^1 = G_8$, and obtain $g_1^* \geq -2.18945802807$. It is easy to test and verify that $G_{g_1^*}^1$ is properly stable if $g_1^* = -2.189458$. In fact, the maximal real part of eigenvalues (a real eigenvalue) ≈ -0.000000088903 for $G_{g_1^*}^1$. According to Hypothesis 1, $G_{g_1^*}^1$ is a properly stable matrix.

4.5 Feedback stabilization of Hessenberg form

Suppose $A = H$, $p = 1$, $B = B_h = (b, 0, \dots, 0)^T$, $K = K_h = (k_1, \dots, k_n)$ in (4.1) and (4.2), where H is a Hessenberg form which can be reduced to from general matrix using stabilized elementary similarity transformations, see [7], and is given by

$$H = \begin{pmatrix} h_{1,1} & h_{1,2} & h_{1,3} & \cdots & h_{1,n-1} & h_{1,n} \\ h_{2,1} & h_{2,2} & h_{2,3} & \cdots & h_{2,n-1} & h_{2,n} \\ & h_{3,2} & h_{3,3} & \cdots & h_{3,n-1} & h_{3,n} \\ & & & \cdots & & \\ & & & & h_{n,n-1} & h_{n,n} \end{pmatrix}. \quad (4.38)$$

Without loss of any generality, we may assume that $h_{i,i-1} \neq 0$ ($i = 2, \dots, n$) in (4.38). Notice that there is no need to commute row/column for reduction to Frobenius-like form from Hessenberg form.

After feedback $u = -K_h x$ of single-input, the state equations $\dot{x}(t) = Hx(t) + B_h u(t)$ become

$$\dot{x} = (H - B_h K_h)x = \begin{pmatrix} h_{1,1} - bk_1 & h_{1,2} - bk_2 & \cdots & h_{1,n} - bk_n \\ h_{2,1} & h_{2,2} & \cdots & h_{2,n} \\ & h_{3,2} & \cdots & h_{3,n} \\ & & \cdots & \cdots \\ & & & h_{n-1,n} \\ & & & h_{n,n} \end{pmatrix} x, \quad (4.39)$$

where $k_i, i = 1, 2, \dots, n$ are regarded as undetermined parameters for stabilization of $H - B_h K_h$. The matrix $H - B_h K_h$ is still a Hessenberg form and its changeable elements are located on the first row.

Proposition 7 shows that the matrix in (4.39) may be stabilized by solving n linear inequalities and solving $n - 2$ inequalities of products of two linear combinations. For the Hessenberg form $H - B_h K_h$, the solution can be found if we investigate firstly n linear inequalities.

Via the stabilized elementary similarity transformations and the scaling strategy, the single row feedback matrix $H - B_h K_h$ in (4.39) is reduced to the Frobenius-like form (4.31). In the computational process, the variables $h_{1,1} - bk_1, h_{1,2} - bk_2, \dots, h_{1,n} - bk_n$ are linearly propagated to the last column of (4.31). We have thus

$$2^{\theta_1} p_{n-1} = 2^{\theta_1} (p_{n-1}^0 - bk_1) < 0, \quad (4.40)$$

$$2^{\theta_2} p_{n-2} = 2^{\theta_2} (p_{n-2}^0 + \gamma_{2,1} k_1 + \gamma_{2,2} k_2) < 0, \dots,$$

$$2^{\theta_s} p_{n-s} = 2^{\theta_s} (p_{n-s}^0 + \sum_{j=1}^s \gamma_{s,j} k_j) < 0,$$

$$2^{\theta_n} p_0 = 2^{\theta_n} (p_0^0 + \sum_{j=1}^n \gamma_{n,j} k_j) < 0,$$

$$(\theta_1 = 0, \theta_2 = \beta_n, \dots, \theta_s = \beta_n + \dots + \beta_{n-s+2}, \dots, \theta_n = \beta_n + \dots + \beta_2),$$

where $p_{n-s}^0 (1 \leq s \leq n)$ and $\gamma_{s,j} (2 \leq s \leq n, 1 \leq j \leq n)$ are given by invariable elements of H . If $\gamma_{s,s} \neq 0 (s = 2, \dots, n)$, then the linear inequalities (4.40) have infinitely many solutions.

Proposition 9 *The Hessenberg matrix $H - B_h K_h$ can be stabilized by the strategies of polynomial, Proposition 3 and Hypothesis 4(i).*

Proof: We first prove that if $\gamma_{s,s} \neq 0$ ($s = 2, \dots, n$) in (4.40), then the Hessenberg matrix $H - B_h K_h$ can be stabilized by the strategies of polynomial, Proposition 3 and Hypothesis 4(i).

In fact, under the assumption, it is easy to see that each k_s ($1 \leq s \leq n$) of (4.40) can be solved successively for any negative real array $(p_{n-1}, \dots, p_0)^T$. Therefore the $n-2$ quadratic inequalities (4.33) can be satisfied if the left-hand side of (4.40)

$$2^{\theta_1} p_{n-1}, 2^{\theta_2} p_{n-2}, \dots, 2^{\theta_{n-1}} p_1, 2^{\theta_n} p_0 \quad (4.41)$$

are, respectively, equal to the coefficients of a stable polynomial or properly stable polynomial. This polynomial is referred to as *polynomial of pole-zero assignment*.

For instance, $-(\lambda + 1)^n$ is such a polynomial of pole-zero assignment, the coefficients (4.41) may be, respectively, equal to the corresponding coefficients of $-(\lambda + 1)^n = -(\lambda^n + n\lambda^{n-1} + \dots + n\lambda + 1)$, namely $p_{n-1} = -n/2^{\theta_1}$, $p_{n-2} = -0.5n(n-1)/2^{\theta_2}$, $\dots$, $p_1 = -n/2^{\theta_{n-1}}$, $p_0 = -1/2^{\theta_n}$. We have thus

$$\begin{aligned} p_{n-1}^0 - bk_1 &= p_{n-1} < 0, p_{n-2}^0 + \gamma_{2,1}k_1 + \gamma_{2,2}k_2 = p_{n-2} < 0, \dots, \\ p_{n-s}^0 + \sum_{j=1}^s \gamma_{s,j}k_j &= p_{n-s} < 0, \dots, p_0^0 + \sum_{j=1}^n \gamma_{n,j}k_j = p_0 < 0, \end{aligned} \quad (4.42)$$

The linear equations (4.42) have unique solution $K_h = (k_1, \dots, k_n)$ for given coefficients (4.41).

Clearly $b \neq 0$ in the linear equations (4.42). Via the algorithm of reduction from Hessenberg matrix to Frobenius-like form, see [1], we see that

$$\gamma_{s,s} = 2^{\beta_2 + \dots + \beta_{n-s+1} - \beta_{s+1} - \dots - \beta_n} \bar{b}_s \neq 0, \quad 2 \leq s \leq n-1, \quad \gamma_{n,n} = \bar{b}_n \neq 0,$$

since, with (4.38), $b_s = -b2^{-\beta_2 - \dots - \beta_s} h_{2,1} \dots h_{s,s-1} \neq 0$, $2 \leq s \leq n$. This complete the proof of the proposition. $\square$

Notice that if the coefficients (4.41) are given by properly stable polynomial of pole-zero assignment, then the stability of stabilized matrix $H - B_h K_h$ is highly sensitive to any perturbation of K_h . Moreover, the solving process is possibly numerically unstable because of $\gamma_{s,s}$. Therefore we should test and verify the stability of $H - B_h K_h$ after K_h is found up.

It would be best if the coefficients (4.41) are given by stable polynomial of pole-zero assignment.

Example 3 Suppose that H_8 is the Hessenberg form reduced to from $A_8 - B_8 K^1$, where A_8 and B_8 are (4.36) and (4.37) respectively, and

$$K^1 = (3.98, 2.05, 2.0, -1.54, -0.34, -0.13, -0.07, 0.29)$$

is the feedback gain selected by (4.35) in the first cycle of single feedback stabilization of Example 2. H_8 is still unstable.

We can stabilize H_8 by Proposition 9. Let $B_h = (4, 0, 0, 0, 0, 0, 0, 0)^T$ and $-(\lambda + 1)^8$ be the polynomial of pole-zero assignment. Solving (4.42) we get

$$K_h \approx (2.30750, 1.07202, 0.44005, 0.51673, 0.38066, 0.09244, -0.04457, 0.85180).$$

It is easy to test and verify the stability of $H_8 - B_h K_h$. Since K_h is an approximate solution of (4.42), the stability of matrix $H_8 - B_h K_h$ has been different from that of the companion matrix of $-(\lambda + 1)^8$.

The matrix $H_8 - B_h K_h$ is stable but not properly stable. We want to make it properly stable matrix by the single feedback stabilization $B_h K_h^1$, where $K_h^1 = (k_1^1, 0, \dots, 0)$. Using (4.32) and (4.33), we find that the selected interval of gain k_1^1 is very small: $-0.000008555 \leq k_1^1 \leq 0.000016050$. It is easy to test and verify that both $H_8 - B_h K_h - B_h K_1^*$ and $H_8 - B_h K_h - B_h K_1^{**}$ are properly stable, where $K_1^* = (-0.000008555, 0, \dots, 0)$ and $K_1^{**} = (-0.000016050, 0, \dots, 0)$ respectively. For $H_8 - B_h K_h - B_h K_1^*$, the maximal real part of eigenvalues (a real eigenvalue) ≈ -0.0000048051 . For $H_8 - B_h K_h - B_h K_1^{**}$, the maximal determining coefficient ≈ 0.465551 . Therefore the properly stable conclusion may be come to from Hypotheses 1 and 4(i) respectively.

We now make $H_8 - B_h K_h$ properly stable matrix by the single feedback stabilization $B_h K_h^2$, where $K_h^2 = (0, k_2^2, 0, \dots, 0)$. Using (4.32) and (4.33), we also find that the selected interval of gain k_2^2 is very small:

$-0.0000217707 \leq k_2^2 \leq 0.0000113129$. It is easy to test and verify that both $H_8 - B_h K_h - B_h K_2^*$ and $H_8 - B_h K_h - B_h K_2^{**}$ are properly stable, where $K_2^* = (0, -0.0000217707, 0, \dots, 0)$ and $K_2^{**} = (0, 0.0000113129, 0, \dots, 0)$ respectively. For $H_8 - B_h K_h - B_h K_2^*$, the maximal determining coefficient ≈ 0.46556914 . For $H_8 - B_h K_h - B_h K_2^{**}$, the maximal real part of eigenvalues (a real eigenvalue) ≈ -0.00000009157 . Therefore the properly stable conclusion may be come to from Hypotheses 4(i) and 1 respectively.

Successively we make $H_8 - B_h K_h$ properly stable matrix by the single feedback stabilizations $B_h K_h^3, \dots, B_h K_h^8$. Similarly we find that all the selected intervals of gains are very small.

These numerical experiments show us that $H_8 - B_h K_h$ is almost a properly stable Hessenberg matrix.

Since a general matrix may be reduced to Hessenberg form, the single row feedback described in this paragraph is easy to stabilize a matrix. That is,

Proposition 10 *Assume that, in the state equation (4.1), A may be reduced to n -order Hessenberg form for any similarity transformation commuting row/column, and there is a column $(b_{1l}, b_{2l}, \dots, b_{nl})^T$ in the control matrix B such that $b_{1l} = 0, \dots, b_{s-1,l} = 0, b_{s,l} = b \neq 0, b_{s+1,l} = 0, \dots, b_{nl} = 0, (1 \leq s \leq n)$. Then we can design the single row feedback matrix*

$$K = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & 0 \\ k_{l1} & k_{l2} & \cdots & k_{ln} \\ 0 & 0 & \cdots & 0 \\ & & \cdots & \\ 0 & 0 & \cdots & 0 \end{pmatrix} \quad (4.43)$$

such that $A - BK$ is stable or properly stable.

Proof: It is easy to see that, via (4.1), $u = -Kx$ and (4.43),

$$A_s^{k_{ij}} = A - BK = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ & & \cdots & \\ a_{s-1,1} & a_{s-1,2} & \cdots & a_{s-1,n} \\ a_{s,1} - bk_{l1} & a_{s,2} - bk_{l2} & \cdots & a_{s,n} - bk_{ln} \\ a_{s+1,1} & a_{s+1,2} & \cdots & a_{s+1,n} \\ & & \cdots & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}.$$

The gain parameters $k_{lj} = (j=1, \dots, n)$ appear only in the s -th row of $A - BK$. Let E_{1s} be the matrix commuting 1- s row/column. Clearly $\bar{A}_1^{k_{ij}} = E_{1s} A_s^{k_{ij}} E_{1s}$ has the same stability as $A_s^{k_{ij}}$. The changeable elements of $A_s^{k_{ij}}$ have been put on the first row of $\bar{A}_1^{k_{ij}}$.

According to the assumption of proposition, $\bar{A}_1^{k_{ij}}$ may be reduced to n -order Hessenberg form by the stabilized elementary similarity transformations. After this reduction is completed, elements of the first row have become

$$\begin{aligned} w_{11} &= a_{ss} - bk_{ls}, w_{12} = h_{12} - b(k_{lj^2} + \sum_{i=3}^n \alpha_{2i} k_{lj^i}), \\ \cdots, w_{1t} &= h_{1t} - b(k_{lj^t} + \sum_{i=t+1}^n \alpha_{ti} k_{lj^i}), \cdots, \\ w_{1,n-1} &= h_{1,n-1} - b(k_{lj^{n-1}} + \alpha_{n-1,n} k_{lj^n}), w_{1n} = h_{1n} - bk_{lj^n}, \end{aligned}$$

where h_{1t} and $\alpha_{ti} (|\alpha_{ti}| \leq 1)$ are given by the elements a_{ij} of A , and $j^2, j^3, \dots, j^n$ is a permutation of $1, \dots, s-1, s+1, \dots, n$.

We now regard $k_{ls}, (k_{lj^2} + \sum_{i=3}^n \alpha_{2i} k_{lj^i}), \dots, (k_{lj^{n-1}} + \alpha_{n-1,n} k_{lj^n}), k_{lj^n}$ as $k_1, k_2, \dots, k_{n-1}$ in (4.39) respectively, and apply Proposition 7 to the Hessenberg form of $\bar{A}_1^{k_{ij}}$. The following equations

$$\begin{aligned} k_{ls} &= k_1^*, k_{lj^2} + \sum_{i=3}^n \alpha_{2i} k_{lj^i} = k_2^*, \cdots, k_{lj^t} + \sum_{i=t+1}^n \alpha_{ti} k_{lj^i} = k_t^*, \cdots, \\ k_{lj^{n-1}} + \alpha_{n-1,n} k_{lj^n} &= k_{n-1}^*, k_{lj^n} = k_n^* \end{aligned} \quad (4.44)$$

have thus been obtained, where $k_1^*, k_2^*, \dots, k_n^*$ is the solution of (4.42).

The linear equations (4.44) have clearly unique solution. $\square$

4.6 Conclusion

Several efficient strategies for feedback stabilization of matrix have been presented via the numerical stabilization of polynomial and matrix.

The basic strategy is to give successively the gains of negative state feedback stabilization with numerical experiments on stability of the Frobenius-like matrix.

Proposition 2 shows that any matrix may be always stabilized properly by diagonal feedback matrix with a suitable gain. If some diagonal elements of matrix aren't allowed to change too big, then the strategies of single feedback stabilization may be considered.

If a matrix is the point-wise or interval controllable canonical form, then Proposition 3, Hypothesis 4, and Proposition 8 are the most efficient stabilized strategies.

Propositions 9 and 10 give efficient stabilized strategies for Hessenberg form and general matrix respectively.

If a matrix is over stable, then it can be made into properly stable matrix by single column feedback stabilization.

References

1. Han JD, Jiang Z, Nie YY (2007) Numerical stabilization of polynomial and matrix. *IMA J. Math. Control & Inform.* 24:473-482
2. Higham Nicholas J (2002) *Accuracy and stability of numerical algorithms*. SIAM, Philadelphia, 2nd ed.
3. Huang L (2003) *Foundations of the theory of stability and robustness*. Chinese Academy of Sciences, Acta Press, Beijing (in Chinese)
4. Kharitonov VL (1978) Asymptotic stability of an equilibrium position of a family of systems of linear differential equation. *Differentsial'nye Uravneniya* 14:1483-1485 (in Russian)
5. Malkin IG (1958) *Theory of motion stability*. Chinese Academy of Sciences, Acta Press, Beijing (translation in Chinese)
6. Nie YY, Nie MH, Yang ZJ (2006) Numerical determination for stability of Hessenberg matrix, dynamics of continuous, discrete and impulsive systems (DCDIS). *Series A Math. Anal.* 13(2):792-803
7. Wilkinson JH (1965) *The algebraic eigenvalue problem*. Oxford University Press, U.K.

Chapter 5

Maxwell geometric dynamics

Constantin Udriște

University Politehnica of Bucharest, Department of Mathematics,
Splaiul Independentei 313, 060042 Bucharest, Romania,
E-mail: udriste@mathem.pub.ro

Abstract. This paper describes the least squares approximations of the solutions of Maxwell PDEs via their Euler-Lagrange prolongations. Section 5.1 recalls the variational problem in electrodynamics. Section 5.2 describes the Ibragimov-Maxwell Lagrangian. Section 5.3 gives the Ibragimov-Udriște-Maxwell Lagrangian and finds its Euler-Lagrange PDEs system. Section 5.4 finds the Euler-Lagrange PDEs system associated to Udriște-Maxwell Lagrangian, and shows that the waves are particular solutions. Section 5.5 studies the discrete Maxwell geometric dynamics. Section 5.6 performs Von Neumann analysis of the associated difference scheme. Section 5.7 addresses an open problem regarding our theory in the context of differential forms on a Riemannian manifold. Section 5.8 underlines the importance of the least squares Lagrangian.

Keywords. Maxwell PDEs, Ibragimov-Maxwell Lagrangian, Udriște-Maxwell Lagrangian, discrete Maxwell geometric dynamics, Von Neumann analysis

5.1 Variational problem in electrodynamics

The classical variational problem in electrodynamics provides a Lagrangian for the wave PDE

$$\Delta A - \frac{1}{c^2} \frac{\partial^2 A}{\partial t^2} = 0$$

regarding the vector potential A of the electromagnetic field, but not for the Maxwell PDEs. This Lagrangian contains the following ingredients:

- the covariant four-vector potential

$$\phi_i = (-cA, \phi);$$

- the electromagnetic tensor field

$$F_{ij} = \frac{\partial \phi_j}{\partial x^i} - \frac{\partial \phi_i}{\partial x^j}$$

or in matrix form

$$(F_{ij}) = \begin{pmatrix} 0 & -cB_z & cB_y & -E_x \\ cB_z & 0 & -cB_x & -E_y \\ -cB_y & cB_x & 0 & -E_z \\ E_x & E_y & E_z & 0 \end{pmatrix};$$

- the Lorentz metric with the diagonal components $(1, 1, 1, -1)$;
- the corresponding contravariant tensor F^{ij} obtained by changing the signs of the elements F_{ij} in the fourth row and fourth column (rising the indices with the Lorentz metric).

All together determine the "least squares" electromagnetic Lagrangian

$$L = -\frac{1}{4} F_{ij} F^{ij}.$$

Udriște [4-15] built a least squares Lagrangian whose Euler-Lagrange PDEs are prolongations of the Maxwell PDEs. Ibragimov [2] built a new kind of Lagrangian just for the Maxwell PDEs.

5.2 Ibragimov-Maxwell Lagrangian

In vacuum, modelled as $R^3 \times R = \{(x, y, z, t)\}$, the Maxwell linear PDEs

$$\frac{1}{c} \frac{\partial E}{\partial t} = \text{curl} H, \quad \text{div} E = 0, \quad \frac{1}{c} \frac{\partial H}{\partial t} = -\text{curl} E, \quad \text{div} H = 0$$

are over-determined. Indeed, they contains six dependent variables, namely, three components of the electric field $E = (E^1, E^2, E^3)$ and three components of the magnetic field $H = (H^1, H^2, H^3)$ and eight equations. On the other hand, it is obvious that the PDEs $\operatorname{div} E = 0$ and $\operatorname{div} H = 0$ hold at any time provided they are satisfied at the initial time $t = 0$. Hence, they can be used as initial conditions.

Let us set $t' = ct$ and take t' as a new t . Also we consider the determined linear PDEs system

$$\operatorname{curl} E + \frac{\partial H}{\partial t} = 0, \quad \operatorname{curl} H - \frac{\partial E}{\partial t} = 0. \quad (5.1)$$

According Ibragimov [2], let us introduce six new dependent variables (Lagrange multipliers) by the vector fields $V = (V^1, V^2, V^3)$ and $W = (W^1, W^2, W^3)$ and the Lagrangian

$$L = V \cdot (\operatorname{curl} E + \frac{\partial H}{\partial t}) + W \cdot (\operatorname{curl} H - \frac{\partial E}{\partial t}). \quad (5.2)$$

Of course, the Euclidean scalar product on R^3 can be replaced with a suitable Riemannian metric. The associated Euler-Lagrange PDEs are

$$\begin{aligned} \frac{\delta L}{\delta V} = \operatorname{curl} E + \frac{\partial H}{\partial t} = 0, \quad \frac{\delta L}{\delta W} = \operatorname{curl} H - \frac{\partial E}{\partial t} = 0 \\ \frac{\delta L}{\delta E} = \operatorname{curl} V + \frac{\partial W}{\partial t} = 0, \quad \frac{\delta L}{\delta H} = \operatorname{curl} W - \frac{\partial V}{\partial t} = 0. \end{aligned}$$

Since these PDEs coincide for $V = E$, $W = H$, the differential operator in PDEs (5.1) is self-adjoint. Consequently, setting $2V = E$, $2W = H$, one obtains the *Ibragimov-Maxwell Lagrangian* [2]

$$L_1 = \frac{1}{2} [E \cdot (\operatorname{curl} E + \frac{\partial H}{\partial t}) + H \cdot (\operatorname{curl} H - \frac{\partial E}{\partial t})]$$

or, in coordinates,

$$\begin{aligned} L_1 = \frac{1}{2} [E^1(E_y^3 - E_z^2 + H_t^1) + E^2(E_z^1 - E_x^3 + H_t^2) + E^3(E_x^2 - E_y^1 + H_t^3) \\ + H^1(H_y^3 - H_z^2 - E_t^1) + H^2(H_z^1 - H_x^3 - E_t^2) + H^3(H_x^2 - H_y^1 - E_t^3)]. \end{aligned}$$

The Euler-Lagrange PDEs produced by the *Ibragimov-Maxwell Lagrangian* L_1 are just the Maxwell PDEs (5.1).

5.3 Ibragimov-Udriște-Maxwell Lagrangian

The Maxwell PDEs system (5.1) and the Euclidean scalar product determine a *least squares Lagrangian* [4-15]. This can be obtained also from (5.2) setting $2V = \text{curl } E + \frac{\partial H}{\partial t}$ and $2W = \text{curl } H - \frac{\partial E}{\partial t}$, i.e.,

$$L_2 = \frac{1}{2}[(\text{curl } E + \frac{\partial H}{\partial t}) \cdot (\text{curl } E + \frac{\partial H}{\partial t}) + (\text{curl } H - \frac{\partial E}{\partial t}) \cdot (\text{curl } H - \frac{\partial E}{\partial t})]$$

or

$$L_2 = \frac{1}{2}[\|\text{curl } E + \frac{\partial H}{\partial t}\|^2 + \|\text{curl } H - \frac{\partial E}{\partial t}\|^2].$$

In coordinates we can write

$$L_2 = \frac{1}{2}[(E_y^3 - E_z^2 + H_t^1)^2 + (E_z^1 - E_x^3 + H_t^2)^2 + (E_x^2 - E_y^1 + H_t^3)^2 + (H_y^3 - H_z^2 - E_t^1)^2 + (H_z^1 - H_x^3 - E_t^2)^2 + (H_x^2 - H_y^1 - E_t^3)^2].$$

The Ibragimov-Udriște-Maxwell Lagrangian L_2 must be accompanied by the PDEs $\text{div } E = 0$ and $\text{div } H = 0$ as initial conditions.

Theorem. *If $\text{div } E = 0$ and $\text{div } H = 0$, then the Euler-Lagrange PDEs associated to L_2 (Euler-Lagrange prolongation of the PDEs system (5.1)) are*

$$\Delta E + \frac{\partial^2 E}{\partial t^2} = 2 \frac{\partial}{\partial t}(\text{curl } H), \quad \Delta H + \frac{\partial^2 H}{\partial t^2} = -2 \frac{\partial}{\partial t}(\text{curl } E).$$

5.4 Udriște-Maxwell Lagrangian

Now let us consider the over-determined Maxwell PDEs system

$$\text{curl } E + \frac{\partial H}{\partial t} = 0, \quad \text{curl } H - \frac{\partial E}{\partial t} = 0, \quad \text{div } E = 0, \quad \text{div } H = 0. \quad (5.3)$$

This PDEs system and the Euclidean scalar product determine the *least squares Udriște-Maxwell Lagrangian* (see [4-15])

$$L_3 = \frac{1}{2} [\| \text{curl } E + \frac{\partial H}{\partial t} \|^2 + \| \text{curl } H - \frac{\partial E}{\partial t} \|^2 + (\text{div } E)^2 + (\text{div } H)^2].$$

The expression of L_3 in coordinates is

$$\begin{aligned} L_3 = \frac{1}{2} [& (E_y^3 - E_z^2 + H_t^1)^2 + (E_z^1 - E_x^3 + H_t^2)^2 + (E_x^2 - E_y^1 + H_t^3)^2 \\ & + (H_y^3 - H_z^2 - E_t^1)^2 + (H_z^1 - H_x^3 - E_t^2)^2 + (H_x^2 - H_y^1 - E_t^3)^2 \\ & + (E_x^1 + E_y^2 + E_z^3)^2 + (H_x^1 + H_y^2 + H_z^3)^2]. \end{aligned}$$

Theorem. *The Euler-Lagrange PDEs associated to L_3 (Euler-Lagrange prolongation of the Maxwell PDEs system (5.3)) are*

$$\Delta E + \frac{\partial^2 E}{\partial t^2} = 2 \frac{\partial}{\partial t} (\text{curl } H), \quad \Delta H + \frac{\partial^2 H}{\partial t^2} = -2 \frac{\partial}{\partial t} (\text{curl } E).$$

Consequently the Euler-Lagrange PDEs of L_2 subject to $\text{div } E = 0$ and $\text{div } H = 0$ are the same with the Euler-Lagrange PDEs of L_3 .

A solution to the electromagnetic PDEs system in the last theorem takes the form of waves $E(r, t) = E_0 e^{i(\omega t - k \cdot r)}$ and $H(r, t) = H_0 e^{i(\omega t - k \cdot r)}$, where t is time (in seconds), ω is the angular frequency (in radians per second), $k = (k_x, k_y, k_z)$ is the wave vector (in radians per meter) and $r = (x, y, z)$. Of course, we accept that the electric field, the magnetic field, and direction of wave propagation are all orthogonal, and the wave propagates in the same direction as $E_0 \times H_0$. The wave vector k is related to the angular frequency by $\|k\| = \frac{\omega}{c} = \frac{2\pi}{\lambda}$, where $\|k\|$ is the wave number and λ is the wave length.

5.5 Discrete Maxwell geometric dynamics

The theory of integrators for Lagrangian dynamics shows that instead of discretization of Euler-Lagrange PDEs we must use a discrete Lagrangian and then discrete Euler-Lagrange equations. Moreover, the discrete Euler-Lagrange equations associated to a least squares Lagrangian can be solved successfully by the Newton method which proves to be convergent for a convenient step (see [10-12]).

The discretization of the Lagrangian $L = L_3$ can be performed by using the *centroid rule* which consists in the substitution of the point (x, y, z, t) with the step $h = (h_1, h_2, h_3, h_4)$, of the point $p = (E^1, E^2, E^3, H^1, H^2, H^3)$ with

$$\frac{p_{klmn} + p_{k+1lmn} + p_{kl+1mn} + p_{klm+1n} + p_{klmn+1}}{5}$$

and of the *partial velocities* p_α , $\alpha = 1, 2, 3, 4$ by

$$\frac{p_{k+1lmn} - p_{klmn}}{h_1}, \quad \frac{p_{kl+1mn} - p_{klmn}}{h_2},$$

$$\frac{p_{klm+1n} - p_{klmn}}{h_3}, \quad \frac{p_{klmn+1} - p_{klmn}}{h_4}.$$

One obtains a discrete Lagrangian

$$L_d : R^4 \times (R^6)^5 \rightarrow R,$$

$$L_d(u_1, u_2, u_3, u_4, u_5) = L(h, \frac{u_1 + u_2 + u_3 + u_4 + u_5}{5}, \frac{u_2 - u_1}{h_1}, \frac{u_3 - u_1}{h_2}, \frac{u_4 - u_1}{h_3}, \frac{u_5 - u_1}{h_4}).$$

This determines the 4-dimensional discrete action

$$S : R^4 \times (R^6)^{(N_1+1)(N_2+1)(N_3+1)(N_4+1)} \rightarrow R,$$

$$S(h, A) = \sum_{k=0}^{N_1-1} \sum_{l=0}^{N_2-1} \sum_{m=0}^{N_3-1} \sum_{n=0}^{N_4-1} L(h, p_{klmn}, p_{k+1lmn}, p_{kl+1mn}, p_{klm+1n}, p_{klmn+1}),$$

where

$$A = (p_{klmn}), \quad k = 0, \dots, N_1, \quad l = 0, \dots, N_2,$$

$$m = 0, \dots, N_3, \quad n = 0, \dots, N_4.$$

We fix the step $h = (h_1, h_2, h_3, h_4)$. The *discrete variational principle* consists in the characterization of the table A for which the action S is stationary, for any family

$$p_{klmn}(\varepsilon) \in R^6$$

with

$$k = 0, \dots, N_1 - 1, \quad l = 0, \dots, N_2 - 1,$$

$$m = 0, \dots, N_3 - 1, \quad n = 0, \dots, N_4 - 1,$$

$$\varepsilon \in I \subset R, \quad 0 \in I, \quad p_{klmn}(0) = p_{klmn}$$

and fixed elements

$$p_{0lmn}, p_{N_1lmn}, p_{k0mn}, p_{kN_2mn}, p_{kl0n}, p_{klN_3n}, p_{klm0}, p_{klmN_4}.$$

The discrete variational principle is obtained using the first order variation of S . In other words the point $A = (p_{klmn})$ is stationary for the action S if and only if

$$\sum_{\xi} \frac{\partial L}{\partial p_{klmn}^i}(\xi) = 0,$$

where ξ runs over five points,

$$\begin{aligned} & (p_{klmn}, p_{k+1lmn}, p_{kl+1mn}, p_{klm+1n}, p_{klmn+1}) \\ & (p_{k-1lmn}, p_{klmn}, p_{k-1l+1mn}, p_{k-1lm+1n}, p_{k-1lmn+1}) \\ & (p_{kl-1mn}, p_{k+1l-1mn}, p_{klmn}, p_{kl-1m+1n}, p_{kl-1mn+1}) \\ & (p_{klm-1n}, p_{k+1lm-1n}, p_{kl+1m-1n}, p_{klmn}, p_{klm-1n+1}) \\ & (p_{klmn-1}, p_{k+1lmn-1}, p_{kl+1mn-1}, p_{klm+1n-1}, p_{klmn}) \end{aligned}$$

and

$$i = 1, \dots, 6; \quad k = 0, \dots, N_1 - 1, \quad l = 0, \dots, N_2 - 1,$$

$$m = 0, \dots, N_3 - 1, \quad n = 0, \dots, N_4 - 1.$$

Giving the elements

$$p_{0lmn}, p_{N_1lmn}, p_{k0mn}, p_{kN_2mn}, p_{kl0n}, p_{klN_3n}, p_{klm0}, p_{klmN_4},$$

the previous algebraic system is solved by the *Newton method* which is contractive for a small step h (see [10-12]).

5.6 Von Neumann analysis

To verify the stability of the previous finite difference scheme, we pass to the *frequency domain*, through what is called *Von Neumann analysis* [1]. For that

1. we accept a uniform grid spacing in all the spatial coordinates, i.e., $h = h_1 = h_2 = h_3$, and an unbounded spatial domain;
2. we denote by $\tau = h_4$ the time step and we define the *space step/time step ratio* to be $v_0 = \frac{h}{\tau}$.

Also, to simplify, we consider the real-valued grid functions of the type

$$E^1 = E^1(x, z, t), \quad E^2 = E^3 = 0$$

$$H^1 = 0, \quad H^2 = H^2(y, z, t), \quad H^3 = 0.$$

The general case $(3+1)D$ is similar and it will be analyzed in a further-coming paper. Also, for the partial derivatives we can use several other forward difference relations or backward difference relations or central difference relations of various orders. Each case could also include error analysis.

Now the least squares Lagrangian L_3 is given by

$$2L_3 = (E_z^1 + H_t^2)^2 + (H_z^2 + E_t^1)^2 + (E_x^1)^2 + (H_y^2)^2.$$

The associated discrete Lagrangian

$$\begin{aligned} 2h^2 L_d = & (p_{klm+1n}^1 - p_{klmn}^1 + (p_{klmn+1}^5 - p_{klmn}^5)v_0)^2 \\ & + (p_{klm+1n}^5 - p_{klmn}^5 + (p_{klmn+1}^1 - p_{klmn}^1)v_0)^2 \\ & + (p_{k+1lmn}^1 - p_{klmn}^1)^2 + (p_{kl+1mn}^5 - p_{klmn}^5)^2, \end{aligned}$$

where $p^1 = E^1$, $p^5 = H^2$, produces the discrete Euler-Lagrange equations

$$\begin{aligned} & 4p_{klmn}^1 - p_{klm+1n}^1 - p_{k+1lmn}^1 - p_{kl-1lmn}^1 - p_{klm-1n}^1 \\ & + (-p_{klm+1n}^5 + p_{klmn}^5 - (p_{klmn+1}^1 - p_{klmn}^1)v_0)v_0 \\ & + (-p_{klmn+1}^5 + p_{klmn}^5 + p_{klm-1n+1}^5 - p_{klm-1n}^5)v_0 \\ & + (p_{klm+1n-1}^5 - p_{klmn-1}^5 + (p_{klmn}^1 - p_{klmn-1}^1)v_0)v_0 = 0 \\ & 4p_{klmn}^5 - p_{klm+1n}^5 - p_{kl+1lmn}^5 - p_{kl-1lmn}^5 - p_{klm-1n}^5 \\ & + (-p_{klm+1n}^1 + p_{klmn}^1 - (p_{klmn+1}^5 - p_{klmn}^5)v_0)v_0 \\ & + (-p_{klmn+1}^1 + p_{klmn}^1 + p_{klm-1n+1}^1 - p_{klm-1n}^1)v_0 \\ & + (p_{klm+1n-1}^1 - p_{klmn-1}^1 + (p_{klmn}^5 - p_{klmn-1}^5)v_0)v_0 = 0. \end{aligned}$$

We need a 3D *discrete spatial Fourier transform* which can be obtained via the substitutions

$$p_{klmn}^1 \rightarrow P_n^1(\alpha, \beta, \gamma)e^{j(\alpha+\beta+\gamma)h}, p_{klmn}^5 \rightarrow P_n^5(\alpha, \beta, \gamma)e^{j(\alpha+\beta+\gamma)h},$$

where (α, β, γ) denotes the *radian spatial wave vector*. We find a system of second order difference equations (*digital filters*)

$$\begin{aligned} & -P_n^1 e^{j\gamma h} + 4P_n^1 - (P_{n+1}^5 - P_n^5)v_0 \\ & + (-P_n^5 e^{j\gamma h} + P_n^5 - (P_{n+1}^1 - P_n^1)v_0)v_0 \\ & - P_n^1 e^{-j\alpha h} - P_n^1 e^{-j\gamma h} + (P_{n+1}^5 - P_n^5)v_0 e^{-j\gamma h} \\ & - (-P_{n-1}^5 e^{j\gamma h} + P_{n-1}^5 - (P_n^1 - P_{n-1}^1)v_0)v_0 = 0, \\ & -P_n^5 e^{j\gamma h} + 4P_n^5 - (P_{n+1}^1 - P_n^1)v_0 \\ & + (-P_n^1 e^{j\gamma h} + P_n^1 - (P_{n+1}^5 - P_n^5)v_0)v_0 \\ & - P_n^5 e^{-j\alpha h} - P_n^5 e^{-j\gamma h} + (P_{n+1}^1 - P_n^1)v_0 e^{-j\gamma h} \\ & - (-P_{n-1}^1 e^{j\gamma h} + P_{n-1}^1 - (P_n^5 - P_{n-1}^5)v_0)v_0 = 0 \end{aligned}$$

that need its stability checked. For this purpose we introduce the *z-transforms* $F^1(z, \alpha, \beta, \gamma)$, $F^5(z, \alpha, \beta, \gamma)$ and we must impose that the *poles* of the recursion do not lie outside the unit circle in the *z-plane*. To simplify, we accept the initial conditions $P_0^1 = 0$, $P_0^5 = 0$. One obtains the homogeneous linear system

$$a_{11}F^1 + a_{12}F^2 = 0, \quad a_{21}F^1 + a_{22}F^2 = 0,$$

where

$$\begin{aligned} a_{11} &= 4z - v_0^2 - v_0^2 z^2 + 2v_0^2 z - 2z \cos(\gamma h) - 2z \cos(\alpha h) \\ a_{22} &= 4z - v_0^2 - v_0^2 z^2 + 2v_0^2 z - 2z \cos(\gamma h) - 2z \cos(\beta h) \\ a_{12} &= a_{21} = -v_0(z^2 - 2z + 1) \\ &+ v_0(z^2 - 2z + 1)\cos(\gamma h) + jv_0 \sin(\gamma h) - jv_0 z^2 \sin(\gamma h). \end{aligned}$$

The poles are the roots of the characteristic equation

$$a_{11}a_{22} - a_{12}^2 = 0,$$

with the unknown z . Explicitly, we have

$$a_4 z^4 + a_3 z^3 + a_2 z^2 + a_1 z + a_0 = 0,$$

where

$$\bar{a}_4 = a_0 = v_0^2(v_0^2 - (e^{j\gamma h} - 1)^2)$$

$$\begin{aligned}
\bar{a}_3 &= a_1 = 2v_0^2(\cos(\alpha h) + \cos(\beta h) + (e^{j\gamma h} - 1)^2 - 2 - 2v_0^2) \\
a_2 &= \bar{a}_2 = -2(2(v_0^2 - 1)\cos^2(\gamma h) \\
&\quad + 2v_0^2(\cos(\alpha h) + \cos(\beta h) - \cos(\gamma h)) - 4v_0^2 - 3v_0^4 - 8 \\
&\quad + 4(\cos(\alpha h) + \cos(\beta h) + 2\cos(\gamma h)) - 2 \sum_{cyclic} \cos(\alpha h)\cos(\beta h)).
\end{aligned}$$

Since in our case $|a_4| = |a_0|$, the Viète relation $z_1 z_2 z_3 z_4 = \frac{a_0}{a_4}$ shows that the condition $|z_i| \leq 1$ is equivalent to $|z_i| = 1, i = 1, 2, 3, 4$. Some general theorems on equations with roots of absolute value unity can be found in [3].

Case 1. For $\text{Im}(a_4) \neq 0$, our equation is a reciprocal complex equation.

Consequently, if z_1 is a root of the equation, then $\frac{1}{\bar{z}_1}$ is also a root.

Therefore, a particular quartic complex reciprocal equation having roots only on unit circle is of the form

$$ae^{-j(\theta_1 + \theta_2)}(z - e^{j\theta_1})^2(z - e^{j\theta_2})^2 = 0, \quad a \in \mathbb{R} \setminus \{0\}, \quad \theta_1, \theta_2 \in [0, 2\pi).$$

Identifying, we obtain the conditions

$$\begin{aligned}
\bar{a}_4 &= a_0 = ae^{j(\theta_1 + \theta_2)}, \bar{a}_3 = a_1 = -2a(e^{j\theta_1} + e^{j\theta_2}) \\
a_2 &= a(e^{j(\theta_1 - \theta_2)} + e^{j(\theta_2 - \theta_1)} + 4).
\end{aligned}$$

If $a = |a_0|$, then the existence of $\theta_1, \theta_2 \in \mathbb{R}$ is equivalent to the system

$$2|a_0| = |a_1|, \quad -8|a_0|a_0 + 4a_2a_0 - a_1^2 = 0.$$

If $a = -|a_0|$, then the existence of $\theta_1, \theta_2 \in \mathbb{R}$ is equivalent to the system

$$2|a_0| = |a_1|, \quad 8|a_0|a_0 + 4a_2a_0 - a_1^2 = 0.$$

These constraints on the *time step* τ , the *grid spacing* h , and *spatial wave vector* (α, β, γ) describe the *stability regions* where our scheme is *marginally stable*.

Case 2. For $\text{Im}(a_4) = 0$ we find $h = \frac{\pi}{\gamma}$ or $h = \frac{2\pi}{\gamma}$ and all the coefficients of the equation are real numbers. To avoid for a moment the complications,

here we discuss only the case when $a_3 = 0, a_2 = 0$, i.e., the case of the biquadratic equation $z^4 + 1 = 0$. The equations $a_3 = 0, a_2 = 0$ lead to

$$\cos(\alpha h), \cos(\beta h) \in \{v_0^2 - 1 \pm \frac{\sqrt{2v_0^2 - 24v_0 + 64}}{2}\}.$$

The condition $\cos(\alpha h) = \cos(\beta h) = 1$ gives the ratio $v_0^2 = \sqrt{68} - 6$; the condition $\cos(\alpha h) = \cos(\beta h) = -1$ produces the ratio $v_0^2 = \sqrt{28} - 2$. All these conditions ensure that our scheme is marginally stable, and over the stability region, the previous constraints give us a time step τ , in terms of the grid spacing h , and implicitly in terms of the wave number γ .

The Von Neumann analysis of the general case will be presented in another paper.

5.7 Open problems

Let us formulate the Udriște-Maxwell theory in terms of differential forms [10]. In this sense, it is well-known that E, H are *differential 1-forms*, J, D, B are *differential 2-forms*, ρ is a *differential 3-form*, and the *star operator* from $D = \varepsilon * E$, $B = \mu * H$ is the *Hodge operator*. If d is the *exterior derivative operator*, and ∂_t is the *time derivative operator*, then the Maxwell's equations for static media are coupled PDEs of first order,

$$dE = -\partial_t B, \quad dH = J + \partial_t D, \quad dD = \rho, \quad dB = 0$$

defined on $R^3 \times R$. Furthermore, some real problems require to replace the *Euclidean manifold* (R^3, δ_{ij}) with a *Riemannian manifold* (R^3, g_{ij}) . In this context the least squares Lagrangian can be written

$$2L_4 = \|dE + \partial_t B\|^2 + \|dH - J - \partial_t D\|^2 + \|dD - \rho\|^2 + \|dB\|^2.$$

Find interpretations for the extremals of L_4 which are not solutions of Maxwell equations. Analyze the influence of the Riemannian metric on the associated discrete Maxwell geometric dynamics. Can we ameliorate the convergence or the stability conditions?

5.8 Conclusions

This paper studies the extremals of the least squares Lagrangian associated to Maxwell PDEs. Between these extremals we have the classical waves, but also other solutions whose interpretation is an open problem. The discrete Maxwell geometric dynamics can produce relevant results since the Newton method for discrete Euler-Lagrange PDEs is contractive in the case of least squares Lagrangian and the Von Neumann analysis proves the stability of the finite difference scheme in reasonable conditions.

Acknowledgements. Partially supported by Grant CNCSIS 86/ 2007 and by 15-th Italian-Romanian Executive Programme of S&T Co-operation for 2006-2008, University Politehnica of Bucharest.

We are very appreciative to Prof. Dr. Nikos Mastorakis, Prof. Dr. Valeriu Prepeliță and Lecturer Romeo Bercia for their suggestions regarding the improvement of numerical aspects for the least squares approximations via Von Neumann analysis.

References

1. Bilbao S (2004) Wave and scattering methods for numerical simulation. John Wiley and Sons, Chichester, UK, p 368
2. Ibragimov NH (2006) Integrating factors, adjoint equations and Lagrangians. *Journal of Mathematical Analysis and Applications* 318(2):742-757
3. Kempner AJ (1935) On the complex roots of algebraic equations. *Bull. Amer. Math. Soc.* 41(12):809-843
4. Udriște C (1999) Nonclassical Lagrangian dynamics and potential maps. In: Conference in Mathematics in Honour of Professor Radu Roșca on the occasion of his Ninetieth Birthday, Katholieke University Brussel, Katholieke University Leuven, Belgium, <http://xxx.lanl.gov/math.DS/0007060> (2000)
5. Udriște C (2000) Solutions of ODEs and PDEs as potential maps using first order Lagrangians. Centenial Vranceanu, Romanian Academy, University of Bucharest, <http://xxx.lanl.gov/math.DS/0007061> (2000); *Balkan Journal of Geometry and Its Applications* (2001) 6(1):93-108
6. Udriște C (2003) Tools of geometric dynamics. *Buletinul Institutului de Geodinamică, Academia Română*, 14(4):1-26; In: Proceedings of the XVIII Workshop on Hadronic Mechanics, honoring the 70-th birthday of Prof. R. M. Santilli, the originator of hadronic mechanics, University of Karlstad, Sweden, June 20-22, 2005; Valer Dvoeglazov, Tepper L. Gill, Peter Rowland, Erick Trell, Horst E. Wilhelm (Eds) Hadronic Press, International Academic Publishers, December 2006, ISBN 1-57485-059-28, pp 1001-1041

7. Udriște C (2003-2004) From integral manifolds and metrics to potential maps. *Atti dell'Accademia Peloritana dei Pericolanti, Classe I di Scienze Fis. Mat. et Nat.* 81-82:1-16, C1A0401008
8. Udriște C (2005) Geodesic motion in a gyroscopic field of forces. *Tensor, N. S.* 66(3):215-228
9. Udriște C (2006) Multi-time maximum principle. Short communication at International Congress of Mathematicians, Madrid
10. Udriște C, Ferrara M, Opriș D (2004) Economic geometric dynamics. *Monographs and Textbooks 6, Geometry Balkan Press*
11. Udriște C, Postolache M (2000) Least squares problem and geometric dynamics. *Italian Journal of Pure and Applied Mathematics* 15:77-88
12. Udriște C, Postolache M, Țevy I (2001) Integrator for Lagrangian dynamics. *Balkan Journal of Geometry and Its Applications* 6(2):109-115
13. Udriște C, Țevy I (2007) Multi-time Euler-Lagrange-Hamilton theory. *WSEAS Transactions on Mathematics* 6(6):701-709
14. Udriște C (2007) Multi-time controllability, observability and bang-bang principle. In: 6th Congress of Romanian Mathematicians, Bucharest, Romania
15. Udriște C, Țevy I (2007) Multi-Time Euler-Lagrange Dynamics. In: *Proceedings of the 7th WSEAS International Conference on Systems Theory and Scientific Computation (ISTASC'07), Athens, Greece, pp 66-71*

Chapter 6

Fractional right ideals

Ulrich Albrecht

Department of Mathematics, Auburn University, Auburn, AL 36849,
U.S.A., E-mail: albreuf@mail.auburn.edu

Abstract. Fractional ideals play a central role in the investigation of integral domains. This paper extends this important concept to right non-singular rings of finite right Goldie-dimension. We show that a satisfactory extension is only possible if R is a semi-prime right and left Goldie-ring.

Keywords. Fractional Right Ideal, Non-Singular Ring, Goldie-Ring

There are several ways to extend the concept of torsion-freeness from modules over integral domains to arbitrary non-commutative rings. The most straightforward approach towards such a generalization is to call a right module M over a ring R *torsion-free in the classical sense* if, for all non-zero $x \in M$ and all regular $c \in R$, one has $xc \neq 0$. Here, $c \in R$ is *regular* if it is neither a right nor a left zero-divisor. To overcome the inherent limitations of this generalization, Goodearl introduced the notion of non-singularity, which is perhaps the most successful extension of torsion-freeness to non-commutative rings (see [6] and [8] for details). The *singular submodule* of a right R -module M is $Z(M) = \{x \in M \mid xI = 0 \text{ for some essential right ideal } I \text{ of } R\}$. The module M is *non-singular* if $Z(M) = 0$ and *singular* if $Z(M) = M$, while the ring R itself is *right non-singular* if R_R is a non-singular module. The right non-singular rings are precisely those rings R which have a regular, right self-injective maximal right ring of quotients which is denoted by Q^r . Furthermore, a submodule U of a right

R -module M is S -closed if M/U is non-singular. A right non-singular ring is a *right Utumi-ring* if every S -closed right ideal of R is a right annihilator. The right and left Utumi-rings are precisely the right and left non-singular rings for which $Q^r = Q^\ell$ [6, Theorem 2.35].

For submodules U and V of Q_R^r , set $(U:V)_\ell = \{q \in Q^r \mid qV \subseteq U\}$ and $(U:V)_r = \{q \in Q^r \mid Vq \subseteq U\}$. Moreover, $\text{ann}_r(S) = \{r \in R \mid Sr = 0\}$ is the right annihilator of a subset S of Q^r . If M is a right (left) R -module, then M^* denotes the left (right) R -module $\text{Hom}_R(M, R)$. A R -module M is *torsionless* if the natural map $\psi_M: M \rightarrow M^{**}$ is a monomorphism. It is *reflexive* if ψ_M is an isomorphism.

An essential submodule U of Q_R^r is a *fractional right ideal* if $IU \subseteq R$ for some essential left ideal I of R . While it is easy to see that the collection of fractional right ideals is closed with respect to finite sums and finite intersections, other of the properties of fractional ideals over an integral domain, which were discussed in [5], may fail to carry over to the more general setting of non-singular rings. For instance, finitely generated submodules of the field of quotients of an integral domain are fractional ideals. However, this need not be the case if R is a right non-singular ring as the next result shows.

Theorem 1. The following are equivalent for a right and left non-singular ring R :

- a) ${}_R R$ is essential in ${}_R Q^r$.
- b) Every essential, finitely generated submodule of Q_R^r is a fractional right ideal.
- c) $(R:U)_\ell$ contains an essential left ideal for all finitely generated submodules U of Q^r .
- d) If U is a fractional right ideal and V is a submodule of Q^r containing U such that V/U is finitely generated, then V is a fractional right ideal.
- e) If U is a fractional right ideal, then so is $U + qU$ for all $q \in Q^r$.

If, in addition R has finite right Goldie dimension, then it satisfies any (and hence all) of the above if and only if it is a right and left Utumi-ring. Moreover, every fractional right ideal U can be embedded into a free module and has the property that U^* is reflexive and a fractional left ideal.

Proof. a) $\Rightarrow$ d): Suppose that U is a fractional right ideal, and select $q_1, \dots, q_n \in Q^r$ such that $V = U + q_1 R + \dots + q_n R$. Since ${}_R R$ is essential in

${}_R Q^r$ by a), there is an essential left ideal I of R with $Iq_i \subseteq R$ for all $i=1, \dots, n$. Moreover, $JU \subseteq R$ for some essential left ideal J of R . Then, $K=I \cap J$ is an essential left ideal R with $KV = KU + Kq_1R + \dots + Kq_nR \subseteq R$.

To see $d) \Rightarrow c)$, let U be a finitely generated submodule of Q^r_R . Since R is a fractional right ideal, $U + R$ is fractional by d). Select an essential left ideal I with $I(U + R) \subseteq R$.

Since $c) \Rightarrow b)$ is clear, we turn to the proof of $b) \text{ or } e) \Rightarrow a)$: In either case, $qR + R$ is a fractional right ideal for all $q \in Q^r$. Consequently, there is an essential left ideal K of R with $Kq \subseteq R$; and Q^r/R is a singular left R -module. Hence, ${}_R R$ is essential in ${}_R Q^r$ provided that Q^r is non-singular as a left R -module. To see this, assume that $Iq = 0$ for some essential left ideal I of R . Select an essential right ideal J of R such that $qJ \subseteq R$. For every $r \in J$, we have $0 = Iqr$. Since R is left non-singular, $qr = 0$. But then, $qJ = 0$ yields $q = 0$ because R is right non-singular.

$a) \Rightarrow e)$: Select an essential left ideal I of R such that $IU \subseteq R$. Since R is an essential in ${}_R Q^r$, I is essential in ${}_R Q^r$, and there is an essential left ideal J of R with $Jq \subseteq I$. Then, $I \cap J$ is an essential left ideal of R with $(I \cap J)(U + qU) \subseteq IU + JqU \subseteq IU \subseteq R$.

Furthermore, if R has finite right Goldie-dimension, then Q^r is semi-simple Artinian by [8, Proposition XIII.3.3]. Then, [6, Proposition 2.27 and Theorem 2.30] guarantee that Q^r is a maximal left ring of quotients of R since ${}_R R$ is essential in ${}_R Q^r$. Therefore, R is a right and left Utumi-ring. Another application of [8, Proposition XIII.3.3] yields that R also has finite left Goldie-dimension.

To show that every fractional right ideal U has the described properties, observe that $M^* \cong (R:M)_\ell / (0:M)_\ell$ for all submodules M of Q^r_R since Q^r_R is right self-injective. Because $(0:M)_\ell = 0$ if M is essential in Q^r_R , we can identify M^* and $(R:M)_\ell$. Since R is a right and left Utumi-ring R , we may also assume $N^* = (R:N)_r$ for all essential submodules N of ${}_R Q$. In particular, $U^* = (R:U)_\ell$. Since U^* therefore contains an essential left ideal, the essential right ideal $U \cap R$ satisfies $U \cap R \subseteq U^{**} = (R:U^*)_r$. Hence, U^* is a fractional left ideal. By symmetry, U^{**} is a fractional right ideal of R , and $U^{***} = (R:U^{**})_\ell \subseteq Q^r$. By the Third Dual Theorem in [7],

U^* is a direct summand of U^{***} . On the other hand, since U^* is a fractional left ideal, it is essential in U^{***} . This is only possible if $U^* = U^{***}$, i.e. U^* is reflexive.

To see that U is isomorphic to a submodule of a free module, it suffices to show that a torsion-less module N of finite Goldie-dimension can be embedded into a finitely generated free module since $U \subseteq U^{**} \subseteq Q^r$. Suppose that N cannot be embedded into a finitely generated free module. Choose an index-set I such that $N \subseteq R^I$, and let $\pi_J: R^I \rightarrow R^J$ be the canonical projection with kernel $R^{I \setminus J}$ whenever J is a subset of I . For $0 \neq x_0 \in N$, select a finite subset J_0 of I with $\pi_{J_0}(x_0) \neq 0$, and consider $U_0 = N \cap R^{I \setminus J_0}$. Since $x_0 \notin U_0$, we have $N/U_0 \neq 0$. Furthermore, N/U_0 is isomorphic to a submodule of R^{J_0} . Consequently, U_0 is an $\mathcal{S}$ -closed submodule of M . Moreover, $U_0 \neq 0$ since otherwise $N \subseteq R^{J_0}$, a contradiction. Choose $0 \neq x_1 \in U_0$ and a finite subset $J_1 \supseteq J_0$ of I with $\pi_{J_1}(x_1) \neq 0$. Then, $U_1 = N \cap R^{I \setminus J_1}$ is a non-zero, $\mathcal{S}$ -closed submodule of N which is contained in U_0 . Inductively, we obtain an infinite descending chain $N \supseteq U_0 \supseteq \dots \supseteq U_n \supseteq \dots$ of $\mathcal{S}$ -closed submodules of N with $U_n \neq U_{n+1}$ for all n . Thus, N has infinite Goldie-dimension, a contradiction.

While a submodule of the field of quotients of an integral domain is fractional if and only if it is isomorphic to an ideal, this does not have to hold in the non-commutative setting, even if R is a finite dimensional right and left Utumi-ring as the next result shows. We want to remind the reader that two submodules U and V of Q^r are *quasi-isomorphic* if there are essential monomorphisms $\alpha: U \rightarrow V$ and $\beta: V \rightarrow U$.

Theorem 2. The following are equivalent for a right non-singular ring R which has finite right Goldie-dimension:

- a) R is a semi-prime right and left Goldie-ring.
- b) i) All finitely generated essential submodules of Q_R^r are fractional right ideals.
 ii) A submodule Q^r is a fractional right ideal if and only if it is isomorphic to an essential right ideal.
- c) Whenever U is a fractional right ideal, then a submodule V of Q_R^r containing U is quasi-isomorphic to U if and only if there is a submodule W of Q_R^r containing V such that W/U is finitely generated.

d) Finitely generated, essential submodules of Q'_R quasi-isomorphic.

Proof. a) $\Rightarrow$ b): The ring R has a semi-simple Artinian classical left and right ring of quotients, Q , which is also its maximal right and left ring of quotients [6, Theorem 3.35]. Hence, all finitely generated essential submodules of Q_R are fractional right ideals by Theorem 1. Moreover, if U is a fractional right ideal, then there exists an essential left ideal I of R with $IU \subseteq R$. By a), there is a regular element $c \in R$ with $c \in I$. Then, cU is a right ideal of R , and $cU \cong U$.

Conversely, assume that U is a submodule of Q_R which is isomorphic to an essential right ideal of R . There is $q \in Q$ such that left multiplication by q is an essential monomorphism from U into R . Consequently, $\dim U = \dim qU = \dim R = \dim Q < \infty$ denotes the Goldie-dimension of an R -module M . Hence, U is an essential submodule of Q_R . Then, $U \cap \text{ann}_r(q) = 0$ yields $\text{ann}_r(q) = 0$. Thus, there are regular elements $c, d \in R$ with $q = d^{-1}c$. Consequently, $cU \subseteq dR \subseteq R$ yields that $(R:U)_\ell$ contains the essential left ideal Rc , proving b).

b) $\Rightarrow$ c): By [6, Lemma 3.33], all regular elements of R are invertible in Q' since R has finite right Goldie-dimension. To see that Q' is the classical left ring of quotients of R , select $q \in Q'$. By i), $U = R + qR$ is a fractional right ideal, for which one can find an essential monomorphism $a:U \rightarrow R$ because of ii). Since Q'_R is injective, a is left multiplication by some $c \in Q'$. Then, $cU \subseteq R$ yields that c is an element of R . It is right regular because $0 = cs = a(s)$ for some $s \in R$ implies $s = 0$ since a is one-to-one. By [6, Lemma 3.33], c is regular and invertible in Q' . If $r = cq \in R$, then $q = c^{-1}r$ by what has been shown. Thus, Q' is the classical left ring of quotients of R . Since Q' is semi-simple Artinian, R is a semi-prime left Goldie-ring. However, a semi-prime right non-singular ring which has finite right Goldie-dimension is right Goldie [6, Corollary 3.32].

Observe that there is a regular element d of R such that dU is an essential right ideal of R since U is a fractional right ideal. If V is quasi-isomorphic to U , then there is a monomorphism $a:V \rightarrow U$. We can find $r, c \in R$ with c regular such that a is left multiplication by $q = c^{-1}r$. Since a is one-to-one, $\text{ann}_r(r) \cap V = 0$. However, V is essential in Q' since it contains U . Thus, r is regular. Hence, $V \subseteq r^{-1}dU \subseteq r^{-1}cd^{-1}R$. Let

$W = U + r^{-1}cd^{-1}R$. Conversely, assume that U and V are contained in a submodule W of Q^r such that W/U is finitely generated. By Theorem 1, W is a fractional right ideal.

c) $\Rightarrow$ d): Let M and N be essential, finitely generated submodules of Q_R^r . Then, $M \cap N \cap R$ is a fractional right ideal. Since $W = M + N$ is a finitely generated submodule of Q^r which contains M and N , we obtain that M and N are quasi-isomorphic to $M \cap N \cap R$ by c).

d) $\Rightarrow$ a): Since R has finite right Goldie-dimension, every essential right ideal I of R contains an essentially finitely generated right ideal J . By d), there is a monomorphism $a: R \rightarrow J$. Clearly, $c = a(1)$ is a right regular element of R which is regular since R has finite right Goldie-dimension. By [6, Theorem 3.35], R is a semi-prime right Goldie ring whose classical right ring of quotients is Q^r .

To show that R is a left Goldie-ring, it remains to verify that it satisfies the left Ore-condition. For $r \in R$ and a regular element c of R , consider $M = R + rc^{-1}R$. By b), there is a monomorphism $\lambda: M \rightarrow R$. We can find $d \in Q^r$ such that $\lambda(x) = dx$ for all $x \in M$. Then, $dM \subseteq R$ yields $dR \subseteq R$, and $d \in R$. Moreover, $0 = dr\lambda(r)$ yields $r = 0$, and d is right regular. However, right regular elements of semi-prime right Goldie-ring are regular. Finally, there is $s \in R$ with $drc^{-1} = s$, i.e. $dr = sc$. Thus, Q^r is also the classical left ring of quotient of R , and R is left Goldie.

However, not much can be said without the finiteness condition on $\dim R$:

Example 1. Let F be a field and I be an infinite set. Consider the ring $R = \bigoplus_I F + (\dots, 1, \dots)F$. Then, R is a commutative non-singular ring whose maximal ring of quotients is $Q = \prod_I F$ [6, Theorem 4.13]. Observe that $J = \bigoplus_I F$ is an essential ideal of Q which is contained in R . Therefore, all submodules U of Q are fractional ideals of R since $JU \subseteq R$. However, Q obviously is not isomorphic to a right ideal of R . Thus, R satisfies condition i), but not condition ii) of part b) of the last theorem. On the other hand, the ring Q satisfies both conditions.

Example 2. Let R be a ring without zero divisors which has infinite left Goldie dimension, but right Goldie dimension 1. The classical right ring of quotients Q_{cl} of R is a division algebra, which also is the maximal right ring of quotients of R [6, Chapter 3.D]. Since Q_{cl} is not the maximal left

ring of quotients of R , ${}_R R$ cannot be essential in ${}_R Q$ [6]. Consequently, R does not satisfy the first condition. Nevertheless, it satisfies the second condition of b): If U is a fractional right ideal of R , then there is a non-zero element q of Q_{cl} with $cU \subseteq R$. Left multiplication by q is a monomorphism from U to R since Q^r is a division algebra.

A ring R is *right fully bounded* if every essential right ideal contains a two-sided ideal which is essential as a right ideal. A submodule of Q_R^r is a *two-sided submodule* of Q^r if it is a submodule of ${}_R Q^r$ too.

Corollary 1. Let R be a right and left fully bounded, semi-prime right and left Goldie-ring. The following are equivalent for a two-sided submodule A of Q_R :

- a) A_R is a fractional right ideal of R .
- b) ${}_R A$ is a fractional left ideal of R .

Proof. By symmetry, it suffices to show that a) implies b). Since A_R is a fractional right ideal of R , there is a regular element c of R such that $cA \subseteq R$. Because R is right fully bounded, the essential right ideal cR contains a two-sided ideal I which is essential as a right ideal. Hence, I contains a regular element d . Then, $Rd \subseteq I \subseteq cR$ yields $A \subseteq c^{-1}R \subseteq Rd^{-1}$. Consequently, $Ad \subseteq R$. Moreover, A is an essential submodule of ${}_R Q$ by [2].

Finally, we show that R cannot always be embedded into a fractional right ideal.

Example 3. There exists a right and left Utumi-ring R of finite Goldie-dimension which contains a two-sided ideal I of R which is essential as a right ideal such that R is not isomorphic to a submodule of I .

Proof. Consider the ring $R = \left\{ \begin{pmatrix} n & x \\ 0 & y \end{pmatrix} \mid n \in \square, x, y \in \square \right\}$. Since the right annihilator of every element of R is generated by one of the idempotents

$$e_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, e_2 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad e_{(x)} = \begin{pmatrix} 0 & x \\ 0 & 1 \end{pmatrix}$$

of R where $x \in \square$, R is right non-singular [4]. Because the additive group of R is a torsion-free abelian group of rank 3, R has finite right and left

Goldie-dimension. However, this guarantees that R is left non-singular too [4, Lemma 8.4]. Furthermore, R is a subring of $Q = Mat_2(\square)$ in such a way that it is essential as a right and as a left submodule. By [6], Q is the maximal right and left ring of quotients of R . Therefore, R is a right and left Utumi-ring.

Finally let $I = \begin{pmatrix} 0 & \square \\ 0 & \square \end{pmatrix}$ which is an essential right ideal of R , and a left ideal of Q . Obviously, R cannot be embedded into I since R has rank 3 as an abelian group while I has rank 2.

References

1. Albrecht U, Dauns J, Fuchs L (2005) Torsion-Freeness and Non-Singularity over Right p.p-Rings. *Journal of Algebra* 285:98-119
2. Albrecht U (2007) Two-sided submodules of Q' . *Houston Journal of Mathematics* 33(1):103-123
3. Anderson F, Fuller K (1992) *Rings and Categories of Modules*. Graduate Texts in Mathematics 13, Springer Verlag
4. Chatters AW, Hajarnavis CR (1980) *Rings with Chain Conditions*. Pitman Advanced Publishing 44, Boston, London, Melbourne
5. Fuchs L, Salce L (2001) *Modules over Non-Noetherian Domains*. AMS 84
6. Goodearl K (1976) *Ring Theory*. Marcel Dekker, New York, Basel
7. Jans J (1963) *Rings and Homology*. Holt, Rinehart, and Winston
8. Stenström B (1975) *Rings of Quotients*. Lecture Notes in Math. 217, Springer Verlag, Berlin, Heidelberg, New York

Chapter 7

A distributed approach for concurrent password recovery of archived files

Esra Celik, Z. Cihan Taysi, A. Gokhan Yavuz

Department of Computer Engineering, Yildiz Technical University,
Besiktas / Istanbul, TURKEY
{esra, cihan, gokhan}@ce.yildiz.edu.tr

Abstract. Compressing files is a common practice for several reasons, such as keeping related files in an archive for easy handling or saving valuable storage space. Furthermore encryption of compressed files is highly encouraged for privacy and security reasons, but loss of encryption password will result in total loss of original data. Also for cryptanalysis purposes one may be required to work with encrypted-compressed data. In this paper, we describe a distributed implementation for concurrent password recovery of archived files in Zip format, which is one of the most popular types of encrypted-compressed archived files. The results obtained from the sample implementation, namely Distributed Password Recovery Platform (DPRP), show that we have achieved a data independent, scalable implementation with optimum domain decomposition and communication overhead.

Keywords. Zip, password recovery, parallel, concurrent, brute force, dictionary, attack

7.1 Introduction

An archive file is a bundle of files packaged together. Separate files are archived because it is easier to handle a single bundled file while moving

in local disk or sending via e-mail. Most archived files are also compressed (shrunk in size) during the archiving process. File compression can be made with a special software utility to reduce the file size along with archiving. The same software must be used to decompress the file before use. Compressed file types can be distinguished by the file extension. The most common types are .ZIP, .TAR, .PKG and .ARJ for PCs; (.SIT for Macintosh). Furthermore encryption of compressed files is highly encouraged for privacy and security reasons, but loss of encryption password will result in total loss of original data [1].

The Zip file format is a popular data compression and archival format. A Zip file contains one or more files that have been compressed, to reduce their file size or stored as-is and can be encrypted with Zip encryption methods. The most popular Zip compression utility WinZip [2] currently uses three types of encryption methods; WinZip 2.0 compatible encryption, 128-bit and 256-bit AES encryption [2]. The newer AES encryption methods are stronger than older WinZip 2.0 compatible encryption methods as they have longer encryption keys. Although extending the encryption key length makes the Zip file safer, the number of different password combinations has still the same value, therefore the performance of a password attack is not directly effected from the key length.

The improvement of today's processors makes it easier to recover passwords in less time than older ones, maybe not years but months. However this time can still be shortened by the way described in this paper.

There are several commercial Zip password recovery tools like Advanced Zip Password Recovery Tool [3], Visual Zip Password Recovery Processor [4], Zip Password Recovery [5] etc. but none of these tools have concurrent operation support.

In this paper we focused on partitioning the overall problem into separate tasks, allocating tasks to processors and synchronizing the tasks to get the desired result on time benefits.

7.2 Encrypted-compressed Zip archive files

A standard encrypted Zip file has 12 bytes header prepended to the compressed data [6]. A checksum approach is used to create this header with three 32-bit keys. Although Adler-32 and CRC-32 (Cyclic Redundancy Check) are both 32-bit checks, the CRC-32 can take on any 32-bit value (2^{32} possibilities), whereas Adler-32 [7] is limited to 65521^2 possibilities [8]. During encryption the key values are initialized using the supplied encryption password. After each byte is encrypted, the keys are then updated

using pseudo-random number generation techniques in combination with the same checksum algorithm [6].

With all printable characters of the ASCII alphabet [9] one can produce 957 different passwords. This possibility is significantly smaller than checksum key combination possibilities. Thus we will focus on solving the problem over a password recovery rather than recovery of checksum keys.

7.2.1 Brute force attack

A brute force attack consists of trying every possible code, combination, or password until the right one is found [10]. Brute force attack is a systematical numeration of all possibilities of a character set. The difficulty of a brute force attack depends on several factors, such as; the length of the password, number of possible values for each digits of the password, and the processing unit's power. Nevertheless a brute force attack will always succeed, eventually. However, brute force attacks against systems with sufficiently long password sizes may require billions of years to complete [10].

7.2.2 Dictionary attack

A dictionary attack consists of trying "every word in a given dictionary" as a possible password for an encrypted message [11]. The dictionary contains possibilities which are most likely to succeed. This method is very useful for recovering passwords that are almost remembered.

There are two methods of improving the success of a dictionary attack. The first method is to use a larger dictionary or more dictionaries. Technical dictionaries and foreign language dictionaries will increase the overall chance of discovering the correct password.

The second method is to perform string manipulation on the dictionary. For example, if the dictionary contains the word "password", common string manipulation techniques will try the word backwards (drowssap), with common number-letter replacements (p4ssw0rd), or with different capitalization (Password) [11].

7.3 Distributed password recovery

One of the first steps in parallelizing a program is to break it into discrete chunks of work that can be distributed to multiple tasks, as known as de-

composition or partitioning. Also the amount of communication is another important issue when parallelizing a problem. Communications frequently impose some type of synchronization between processes, which causes processes to spend more time on waiting than doing work. Data dependency should also be given attention as it is one of the primary inhibitors to parallelism. The idle time of processes must be minimized, thus they must be kept as busy as possible doing their assigned work which infers a load balanced solution to a given problem [18].

Our distributed approach for concurrent password recovery of archived files which is called Distributed Password Recovery Platform (DPRP) surely considered the design issues which are mentioned above. The sample implementation of DPRP is done on a Beowulf Cluster [16].

DPRP partitions the password recovery problem by scattering the total range of possible passwords to the nodes where each node runs an exact copy of password recovery process which is shown in Figure 7.1.

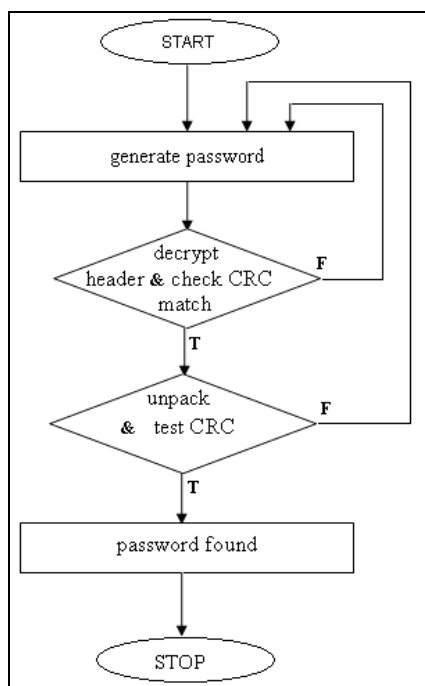


Fig. 7.1. Main outline of the password recovery process

All DPRP processes running across the nodes generate and test the correctness of the passwords independently from other processes. In DPRP, communication overhead is significantly lower than the processing

overhead as DPRP processes only need to communicate to notify each other if the correct password is found.

Once a DPRP process starts iterating over its assigned range of possible passwords, it does not require any data from other processes, thus a maximum degree of data dependency is achieved.

Data independency and subtle communication of DPRP processes make DPRP an excellent candidate for load balancing and scalability. Increased number of DPRP nodes will yield decreased recovery time.

As it is seen above, Zip password recovery is clearly convenient for parallelization. Now we will focus on how we partitioned the data and implemented the brute force and dictionary attack methods.

At first, all processes determine their own start and end point of brute force or dictionary attacks. Each of them generates and tries candidate passwords which are in their own range. While processes try candidate passwords sequentially, they test if any other process has found the correct password. It is known that messaging decreases the performance and should not be done when it is not very necessary or when it can be avoided. In DPRP in order to minimize messaging overhead, processes does not call the test message function in every cycle. This check is done only after a dynamically determined number of cycles. This number will be calculated over the current length of the generated password.

7.3.1 Achieving data independency for brute force attack

When DPRP processes run in brute force attack mode, they themselves partition the given set of characters into equal chunks and each process generates candidate passwords from the global set of characters starting with the characters contained in their own chunk.

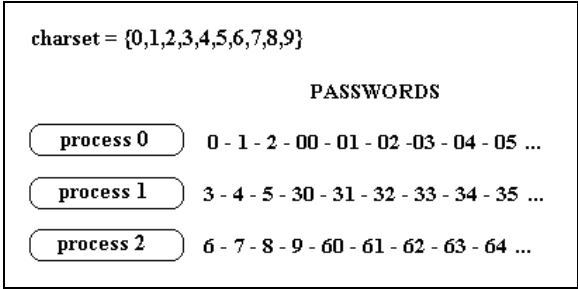


Fig. 7.2. Password generation

Figure 7.2 outlines the decomposition of a given character set into chunks and the generation of candidate passwords. Pseudo code for this operation is given in below.

1. Calculate number of characters in chunks
2. Calculate the start point by using process id and chunk size
3. Calculate the end point by using start point and chunk size
4. Generate the first candidate password

Splitting the data with this method also increases the possibility to reach the correct password in less time (Figure 7.3).

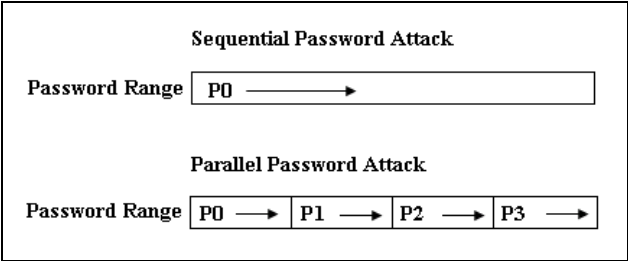


Fig. 7.3. Partitioning the candidate passwords into chunks

7.3.2 Achieving data independency for dictionary attack

In dictionary attack mode, each DPRP process uses candidate passwords within its calculated part of a common dictionary file which resides on a shared storage. Pseudo code for processes to calculate the password range and get the first password from a dictionary is given below.

1. Get number of passwords in dictionary
2. Calculate the starting line for process by using process id
3. Calculate the number of candidate password for each process
4. Get passwords line by line

Each process opens the dictionary file, finds their first and total number of lines, gets and tries the passwords in their own range (Figure 7.4).

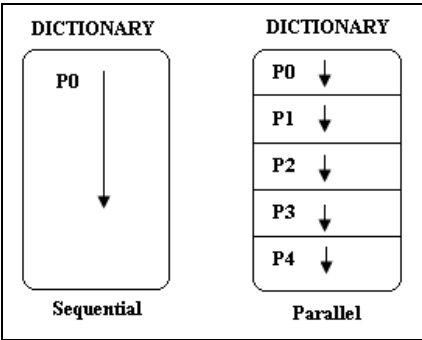


Fig. 7.4. Partitioning the dictionary for processes

7.3.3 Coding and performance issues

C programming language and LAM/MPI [12] library were used to implement the DPRP. In order to increase the concurrency of DPRP processes, each DPRP process makes use of non-blocking MPI_Irecv [13] call to check whether another DPRP process has found the correct password. MPI_Irecv function allocates a communication request object and associates it with a request handle, which means the status of the communication, must be queried using MPI_Test [13] function to get notified about the result of the operation. As excessive use of MPI_Test will diminish the positive affect of the MPI_Irecv usage, we minimized the usage of the MPI_Test, by checking the status of other DPRP process once only after a dynamically determined number of cycles. This behaviour can be considered analogous trying to keep a pipeline as full as possible.

7.4 Conclusion

DPRP is a distributed approach for concurrent password recovery of archive files, and it is implemented using C and LAM/MPI library on a Beowulf Cluster. Experimental results obtained from the sample implementation of DPRP are very promising and show that we have achieved a data independent, scalable implementation with optimum domain decomposition and communication overhead.

Figure 7.5 and Figure 7.6 clearly shows how well DPRP is scaled, as the number of passwords that can be generated per second increases promotional to the increased number of cooperating DPRP processes.

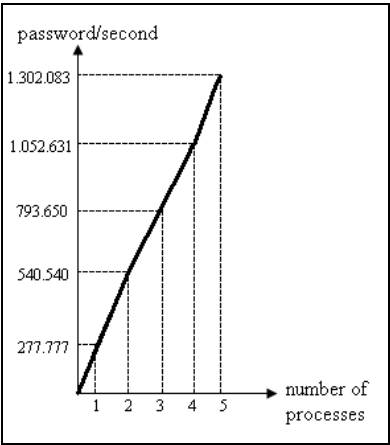


Fig. 7.5. Number of passwords generated and test per second

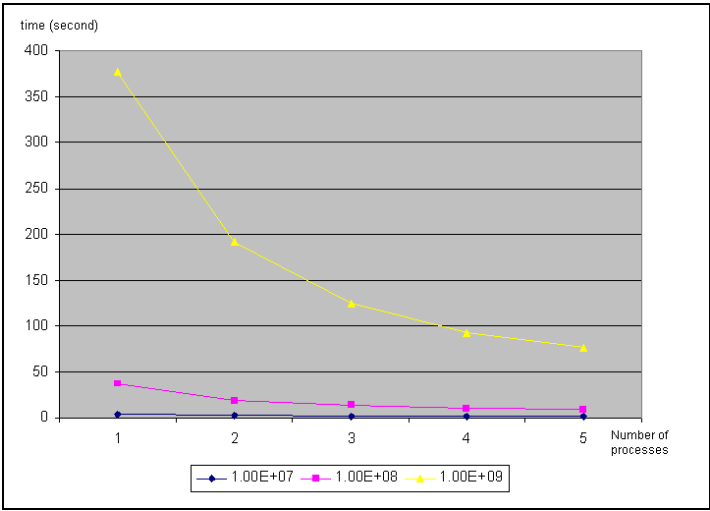


Fig. 7.6. Required time to generate and test candidate passwords

Although in the sample implementation of DPRP we have focused on standard Zip encryption method with CRC-32 and Adler-32 checksum algorithms, DPRP can be applied to other encrypted-compressed data formats.

It is important to note that trying to recover the encryption keys [15] directly will be better than password recovery if the password length is more than 14 digits.

DPRP currently runs on Beehive Cluster [14] which has five nodes with 2 GB of memory and two Intel P3 700 Mhz processors. The performance

of the application can be increased by increasing the number of nodes or improving the processors.

References

1. In a Nutshell. <http://www.telacomunications.com/nutshell/achive.htm>
2. WinZip Computing, Inc. Homepage (2007) <http://www.winzip.com/>
3. Advance Zip Password Recovery Homepage, <http://www.elcomsoft.com/azpr.html>
4. Visual Zip Password Recovery Processor Homepage, <http://www.zipcure.com/>
5. Atomic Zip Password Recovery Homepage, <http://www.apasscracker.com/products/zip.php>
6. ZIP File Format Specification (2007) Version 6.3.1, <http://www.pkware.com/documents/casestudies/APPNOTE.TXT>
7. Deutsch P (1996) ZLIB Compressed Data Format Specification version 3.3. <http://www.ietf.org/rfc/rfc1950.txt>
8. ZLIB Software Web Page (2006) http://www.zlib.net/zlib_tech.html
9. ASCII Table Web Page, <http://www.asciitable.com/>
10. What is a brute force attack? Cryptology FAQs, <http://www.tech-faq.com/brute-force-attack.shtml>
11. What is a dictionary attack? Cryptology FAQs, <http://www.tech-faq.com/dictionary-attack.shtml>
12. The LAM/MPI Project Homepage (2007) Available at <http://www.lam-mpi.org>
13. Rance Necaise Department of Computer Science (2007) <http://www.cs.wlu.edu/~necaise/ParallelComputing/MPIFunctionReference>
14. Beehive Clustering System Homepage. <http://beehive.yildiz.edu.tr>
15. Stevens M, Flanders E Available at <http://archive.cert.uni-stuttgart.de/vuln-dev/2003/02/msg00019.html>
16. The Beowulf Cluster Site (2007) Available at <http://www.beowulf.org/>
17. Stokes J (2000) SIMD Architectures. <http://arstechnica.com/articles/paedia/cpu/simd.ars>
18. Barney B (2006) Introduction to Parallel Computing. http://www.llnl.gov/computing/tutorials/parallel_comp/#DesignPartitioning

Chapter 8

Solutions for an integral-differential equation

Nikos E. Mastorakis,¹ Olga Martin²

¹ Military Institutes of University Education
Hellenic Naval Academy, Greece.

<http://www.wseas.org/mastorakis>

² University “Politehnica” of Bucharest
313, Splaiul Independentei, Romania
omartin_ro@yahoo.com

Abstract. In this paper we find a continuous, non-constant and periodical solution for an integral-differential equation with the variable coefficients. This is compared with the solution obtained with the help of the finite differences method.

Keywords. Neutron transport equation, method of finite differences, the trapezoidal rule, series uniformly convergent

8.1 Introduction

The main problem in the nuclear reactor theory is to find the neutrons distribution in the reactor, hence their density. This is the solution of an integral-differential equation named the neutron transport equation. Many authors paid attention to this problem and its applications [1, 3, 9, 11], but an exact solution for this equation there is only for any particular cases.

Using the theory of the integral depending on parameter and Cauchy principal value of the improper integral involving unbounded functions [4, 12], we provide a periodical analytical solution for the one-dimensional stationary problem (8.1), where φ is the neutron density and g is the source

function. The neutron moves to a direction, which makes the v angle with Ox axis and $y = \cos v$. We prove that, if the function g is the sum of a uniformly convergent series, then the solution φ is the sum of a uniformly convergent series, too.

With the method of the finite differences and the trapezoidal approximation [2, 5-6, 10], we obtain a numerical solution for our problem and this is compared with the analytical solution.

8.2 Problem formulation

Let us now consider the integral-differential equation of the neutron transport theory:

$$y \frac{\partial \varphi(x, y)}{\partial x} = \int_{-1}^1 \varphi(x, \xi) d\xi + g(x, y) \quad (8.1)$$

where u is a continuous and non-constant function defined on the domain $D = [0, H] \times [-1, 1]$. We find the form of function $g(x, y)$ so that, the solution u to be periodical function with respect to the variable x . If the period is $H \in \mathbf{Z}$, then we have

$$\varphi(x, y) = \varphi(x + H, y), \quad \forall y \in [-1, 1] \quad (8.2)$$

The equation (8.1) may be written in the form

$$y \frac{\partial \varphi(x, y)}{\partial x} = v(x) + g(x, y) \quad (8.3)$$

where

$$v(x) = \int_{-1}^1 \varphi(x, \xi) d\xi \quad (8.4)$$

Integrating (8.3) with respect to the variable x we obtain for $\forall x \in [0, H]$:

$$u(x, y) = \frac{1}{y} \int_0^x v(\xi) d\xi + \frac{1}{y} \int_0^x g(\xi, y) d\xi + c(y) \quad (8.5)$$

Now we get

$$v(x) = \int_{-1}^1 c(y) dy + \int_0^x v(\xi) d\xi \int_{-1}^1 \frac{1}{y} dy + \int_{-1}^1 \frac{1}{y} \left(\int_0^x g(\xi, y) d\xi \right) dy \quad (8.6)$$

The improper integral, $\int_{-1}^1 \frac{dy}{y}$, diverges, but there exists its Cauchy principal value given by the formula

$$v.p. \int_{-1}^1 \frac{dy}{y} = \lim_{\varepsilon \rightarrow 0} \left(\int_{-1}^{-\varepsilon} \frac{dy}{y} + \int_{\varepsilon}^1 \frac{dy}{y} \right) = 0$$

and
$$v(x) = \int_{-1}^1 c(y) dy + \int_{-1}^1 \frac{1}{y} \left(\int_0^x g(\xi, y) d\xi \right) dy$$

Replacing $v(x)$ in (8.5) we have

$$\varphi(x, y) = c(y) + \frac{x}{y} \int_{-1}^1 c(y) dy + \frac{1}{y} \int_0^x g(\xi, y) d\xi + \frac{1}{y} \int_0^x \left(\int_{-1}^1 \frac{1}{y} \left(\int_0^\xi g(t, y) dt \right) dy \right) d\xi \quad (8.7)$$

and because $u(x, y)$ is a continuous function on D , we choose $c(y)$ so that

$$\int_{-1}^1 c(y) dy = 0 \quad (8.8)$$

This means that $c(y)$ may be of the form

$$c(y) = D_1 y^{2n-1}, \quad n = 1, 2, \dots \quad (8.9)$$

or

$$c(y) = D_2 \sin(2\pi n y), \quad n \in \mathbf{Z} \quad (8.10)$$

or

$$c(y) = D_1 \cos(\pi n y), \quad n \in \mathbf{Z} \quad (8.11)$$

where D_i are constants. Let us now consider

$$c(y) = y \quad (8.12)$$

and let

$$g(x, y) = \sum_{k=1}^{\infty} (A_k(y) \cos(2k\pi x) + B_k(y) \sin(2k\pi x)) = \sum_{k=1}^{\infty} g_k(x, y) \quad (8.13)$$

Then

$$\varphi(x, y) = \sum_{k=1}^{\infty} \varphi_k(x, y) + c(y)$$

For a fixed y , the series (8.13) is uniformly convergent on any interval $(0, x)$, therefore the series can be integrated term wise from zero to x and the next integral from (8.7) becomes:

$$\begin{aligned} \frac{1}{y} \int_0^x g_k(t, y) dt &= \frac{1}{y} (A_k(y) \frac{\sin(2k\pi t)}{2k\pi} \Big|_{t=0}^{t=\xi} - \frac{B_k(y)}{2k\pi} \cos(2k\pi t) \Big|_{t=0}^{t=\xi}) = \\ &= \frac{A_k(y) \sin(2k\pi \xi)}{2k\pi y} + \frac{B_k(y) \sin^2(k\pi \xi)}{k\pi y} \end{aligned}$$

The last term from (8.7) will be of the form

$$\begin{aligned} &\frac{1}{y} \int_0^x \left[\int_{-1}^1 \frac{1}{s} \left(\int_0^\xi (A_k(s) \cos 2k\pi t + B_k(s) \sin 2k\pi t) dt \right) ds \right] d\xi = \\ &= \frac{\sin^2 k\pi x}{2k^2 \pi^2 y} \int_{-1}^1 \frac{A_k(s)}{s} ds + \frac{1}{y} \left(\frac{x}{2k\pi} - \frac{\sin 2k\pi x}{4k^2 \pi^2} \right) \int_{-1}^1 \frac{B_k(s)}{s} ds \end{aligned} \quad (8.14)$$

and the continuity condition of φ on D leads to

$$\int_{-1}^1 \frac{B_k(s)}{s} ds = 0. \quad (8.15)$$

Then, we obtain for each index k :

$$\begin{aligned} \varphi_k(x, y) &= \frac{1}{y} \int_0^x (A_k(y) \cos 2k\pi \xi + B_k(y) \sin 2k\pi \xi) d\xi + \\ &+ \frac{\sin^2 k\pi x}{2k^2 \pi^2 y} \int_{-1}^1 \frac{A_k(s)}{s} ds = \frac{A_k(y) \sin 2k\pi x}{2k\pi y} + \\ &+ \left(\frac{B_k(y)}{k\pi y} + \frac{1}{2k^2 \pi^2 y} \int_{-1}^1 \frac{A_k(s)}{s} ds \right) \cdot \sin^2 k\pi x \end{aligned} \quad (8.16)$$

Also, in view of the continuity of the $\varphi_k(x, y)$ in any point $y \in [-1; 1]$, it is necessary that

$$\frac{B_k(y)}{k\pi y} + \frac{1}{2k^2 \pi^2 y} \int_{-1}^1 \frac{A_k(s)}{s} ds = 0.$$

Let us now consider the functions A_k of the form

$$A_k(y) = y^k, \quad k \geq 1 \quad (8.17)$$

and then $B_k(y)$ becomes

$$B_k(y) = -\frac{1}{2k\pi} \int_{-1}^1 \frac{A_k(s)}{s} ds \Rightarrow B_k(y) = \begin{cases} 0, & \text{if } k \text{ even} \\ -\frac{1}{k^2\pi}, & \text{if } k \text{ odd} \end{cases} \quad (8.18)$$

Finally, from (8.7), (8.12), (8.16), (8.17) and (8.18) we obtain

$$\varphi(x, y) = y + \sum_{k=1}^{\infty} y^{2k-2} \cdot \frac{\sin 2(2k-1)\pi x}{2(2k-1)\pi} \quad (8.19)$$

for

$$g(x, y) = \sum_{k=1}^{\infty} \left(y^{2k-1} \cos 2(2k-1)\pi x - \frac{\sin 2(2k-1)\pi x}{(2k-1)^2 \pi} \right). \quad (8.20)$$

We now prove that the functional series (19) is uniformly convergent on $(0;1)$. Let us consider $H = 1$ and y fixed in $[-1; 1]$. We remind the Dirichlet's test:

"If series $\sum_{n=1}^{\infty} f_n(x)$, $f_n : D \subset \mathbf{R} \rightarrow \mathbf{R}$ is of the form $\sum_{n=1}^{\infty} \alpha_n(x) v_n(x)$, the sequence $\{S_n(x)\}$ of the partial sums for the series $\sum_{n=1}^{\infty} v_n(x)$ is equal bounded on $B \subset D$ and $\{\alpha_n(x)\}$ is a sequence which is uniformly convergent to zero on B , then it is uniformly convergent on B ."

Let

$$\begin{aligned} S_n(x) &= \sum_{k=1}^n \sin 2(2k-1)\pi x \Rightarrow \\ |\sin 2\pi x + \sin 6\pi x + \sin 10\pi x + \dots + \sin 2(2n-1)\pi x| &= \\ &= \frac{\sin^2(2n\pi x)}{|\sin 2\pi x|} \leq \frac{1}{|\sin 2\pi x|}, \end{aligned}$$

$\forall x \in \left[\frac{\varepsilon}{2\pi}, 1 - \frac{\varepsilon}{2\pi} \right] = B \subset [0;1]$, where $\varepsilon > 0$ is very small in comparison with the unity and n a natural number. Then, exists $M = 1/\sin \varepsilon > 0$, such that $|S_n(x)| < M$, $\forall n \in \mathbf{N}$, $\forall x \in \left[\frac{\varepsilon}{2\pi}, 1 - \frac{\varepsilon}{2\pi} \right] \subset [0;1]$.

- If y is fixed, $y \in [-1; 1]$, $\left\{ \frac{y^{2(k-1)}}{2k-1} \right\}_k$, $k \in \mathbf{N}^*$, is a sequence which converges to zero and $\sum_{k=1}^n \sin 2(2k-1)\pi x$ is equal bounded series. It results from Dirichlet's test, that series (8.19) is uniformly convergent on B .
- Let us now consider in (8.19), x fixed. The numbers series $\sum_{k=1}^{\infty} \sin 2(2k-1)\pi x$ has the sequence of the partial sums bounded by $\frac{1}{|\sin 2\pi x|}$, $\forall x \in \left[\frac{\varepsilon}{2\pi}, 1 - \frac{\varepsilon}{2\pi} \right] \subset [0; 1]$ and the sequences of functions, $\left\{ \frac{y^{2(k-1)}}{2k-1} \right\}_k$, $k \in \mathbf{N}^*$ are uniformly convergent to zero on $[-1; 1]$.

Thus, the uniform convergence of the functional series (8.19) is proved, too.

Using (8.20), the source function, $g(x, y)$ must to be periodically with respect to the variable x and the following result can be derived.

Theorem

Let φ a non-constant and continuous solution on D for one-dimensional transport equation (8.1). If the source function, $g(x, y)$ is a periodic function with respect to x , with $H = 1$, of the form

$$g(x, y) = \sum_{k=1}^{\infty} \left(y^{2k-1} \cos 2(2k-1)\pi x - \frac{\sin 2(2k-1)\pi x}{(2k-1)^2 \pi} \right) \quad (8.20)$$

then the following function

$$\varphi(x, y) = y + \sum_{k=1}^{\infty} y^{2k-2} \cdot \frac{\sin 2(2k-1)\pi x}{2(2k-1)\pi} \quad (8.19)$$

is a periodic solution with respect to x of the equation (8.1).

■ Using (8.19), the solution φ is expressed as a uniform convergence series. Since all terms of this series are continuous with continuous derivatives and the series of derivatives converges uniformly on D (see the prove for φ), it can be integrated and differentiated term-wise. Taking g and φ of the form (8.19) and (8.20), the equation is immediately verified. ■

8.3 Numerical method

Further, to find the solution of the problem (8.1) – (8.2) a numerical algorithm is presented. Let a domain $D=[0,H]\times[-1,1]$ be, $H=1$ and a grid:

$$\Delta_D = \{(x_i, y_j) | x_{i+1} = x_i + h, y_{j+1} = y_j + \tau\}$$

where $i = 0, 1, \dots, N, j = 0, 1, \dots, M-1, x_0 = 0, x_{N+1} = 1, y_0 = -1, y_M = 1$. In the network we have: $N = 2, M = 5, h = 1/3$ and $\tau = 2/5$ and the coordinates of the nodes $P(x_i, y_j) = \mathbf{ij}$ are presented in the Figure 8.2.

The function g is calculated by (8.20) (summation index $k = 1$):

$$g(x_i, y_j) = y_j \cos 2\pi x_i - \frac{\sin 2\pi x_i}{\pi}$$

and the surface $z = g(x, y)$ is presented in the Figure 8.1.

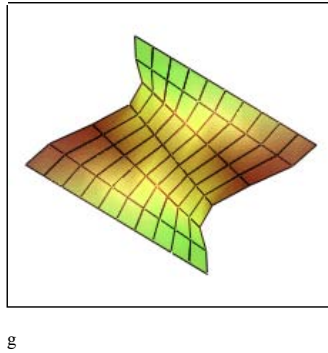


Fig. 8.1. The surface $z = g(x, y)$

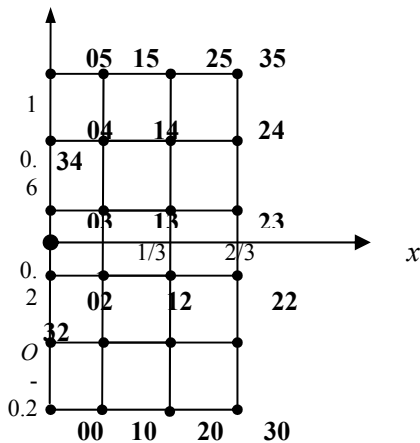


Fig. 8.2. The coordinates of the nodes $P(x_i, y_j) = \mathbf{ij}$

Using the method of finite difference and the trapezoidal rule for the integral calculus we obtain a system of linear equations, which depends of N and M .

1. If N is an even natural number we have

- for $1 \leq i \leq N/2$:

$$y_k \cdot \frac{\varphi(x_{i-1}, y_k) - \varphi(x_{i+1}, y_k)}{2h} - \tau \cdot \left(\frac{\varphi(x_i, y_0)}{2} + \varphi(x_i, y_1) + \dots + \varphi(x_i, y_{M-1}) + \frac{\varphi(x_i, y_M)}{2} \right) = g(x_i, y_k)$$

- for $N \geq i \geq N/2 + 2$

$$y_k \cdot \frac{\varphi(x_{i+1}, y_k) - \varphi(x_{i-1}, y_k)}{2h} - \tau \cdot \left[\frac{\varphi(x_i, y_0)}{2} + \dots + \frac{\varphi(x_i, y_M)}{2} \right] = g(x_i, y_k)$$

and the similar relations:

$$\begin{aligned} & y_k \cdot \frac{\varphi(x_{N/2-1}, y_k) - \varphi(x_{N/2+1}, y_k)}{2h} - \\ & - \tau \cdot \left[\frac{\varphi(x_{N/2}, y_0)}{2} + \dots + \frac{\varphi(x_{N/2}, y_M)}{2} \right] = g(x_{N/2}, y_k) \\ & y_k \cdot \frac{\varphi(x_{N/2+2}, y_k) - \varphi(x_{N/2}, y_k)}{2h} - \\ & - \tau \cdot \left[\frac{\varphi(x_{N/2+1}, y_0)}{2} + \dots + \frac{\varphi(x_{N/2+1}, y_M)}{2} \right] = g(x_{N/2+1}, y_k) \end{aligned}$$

$k \in 0, 1, \dots, M.$

2. If N is an odd natural number we have

- for $1 \leq i < (N+1)/2$:

$$y_k \cdot \frac{\varphi(x_{i-1}, y_k) - \varphi(x_{i+1}, y_k)}{2h} - \tau \cdot \left(\frac{\varphi(x_i, y_0)}{2} + \varphi(x_i, y_1) + \dots + \varphi(x_i, y_{M-1}) + \frac{\varphi(x_i, y_M)}{2} \right) = g(x_i, y_k)$$

- for $(N+1)/2 < i \leq N$:

$$\begin{aligned} & y_k \cdot \frac{\varphi(x_{i+1}, y_k) - \varphi(x_{i-1}, y_k)}{2h} - \\ & - \tau \cdot \left[\frac{\varphi(x_i, y_0)}{2} + \dots + \frac{\varphi(x_i, y_M)}{2} \right] = g(x_i, y_k) \end{aligned}$$

- and for $i = (N+1)/2$:

$$y_k \cdot \frac{\varphi(x_{i+1}, y_k) - \varphi(x_{i-1}, y_k)}{2h} -$$

$$-\tau \cdot \left[\frac{\varphi(x_i, y_0)}{2} + \varphi(x_i, y_1) + \dots \right] = g(x_i, y_k), k = 0, 1, \dots, M.$$

We add at the systems of 12 linear equations in 12 unknowns that corresponding to **1** or **2**, the boundary conditions:

$$\varphi(x_0, y_k) = \varphi(x_{N+1}, y_k) = y_k, \quad k = 0, 1, \dots, M,$$

$$\varphi(x_i, 1) = 1 + \frac{\sin 2\pi x_i}{2\pi};$$

$$\varphi(x_i, -1) = -1 + \frac{\sin 2\pi x_i}{2\pi}, \quad i = 1, 2, \dots, N.$$

The approximate values of the solution are compared with the values of the analytical solution and the results there are in the Table 8.1:

Table 8.1. Comparison between the approximate values and the values of the analytical solution

	$\varphi_{\text{analytical}}$	$\varphi_{\text{numerical}}$
φ_{10}	-0.8	-0.6
φ_{20}	-1.1	-0.8
φ_{11}	-0.46	-0.32
φ_{21}	-0.7	-0.5
φ_{12}	-0.06	-0.1
φ_{22}	-0.3	-0.2
φ_{13}	0.3	0.2
φ_{23}	0.06	0.1
φ_{14}	0.7	0.5
φ_{24}	0.46	0.32
φ_{15}	1.1	0.8
φ_{25}	0.8	0.6

In the Figure 8.3 are presented the surfaces that correspond to the exact solution, $\varphi_{\text{analytical}}$ (red color) and the numerical solution, $\varphi_{\text{numerical}}$ (green color).

8.4 Conclusions

The comparative study of the analytical and numerical periodical solution of (8.1) shows that the errors of the approximate solution decrease if in the series (8.20) of g , we consider $k \geq 2$. Also, we conclude from the numerical integration method that the values of the absolute errors in the presented algorithm are of the h^2 and τ^2 orders.

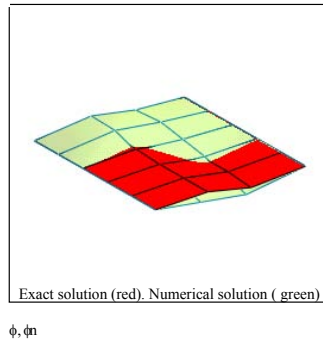


Fig. 8.3. The surfaces that correspond to the exact solution

References

1. Ackroyd RT, Gashut AM, OA Abuzid O (1996) Discontinuous variational solutions for the neutron diffusion equation. *Ann. Nucl. Energy* 23:1289-1294
2. Case KM, Zweifel PF (1967) *Linear transport theory*. Addison-Wesley, Massachusetts
3. Davis WR (1970) *Classical fields, particles and the theory of relativity*. Gordon and Breach, New York
4. Glasstone S, Kilton C (1982) *The elements of nuclear reactors theory*. Van Nostrand, Toronto – New York – London
5. Marchouk G (1980) *Méthodes de Calcul Numérique*, Édition MIR de Moscou
6. Marchouk G, Shaydourov V (1983) *Raffinement des Solutions des Schémas aux Différences*. Édition MIR de Moscou
7. Marciuk G, Lebedev V (1971) *Cislennie Metodi v Teorii Perenosa Neitronov*. Atomizdat, Moscova
8. Martin O (1992) Une Méthode de Résolution de l'Équation du Transfert des Neutrons. *Rev.Roum. Sci. Tech.- Méc. Appl.* 37:623-646
9. Mastorakis N, Martin O (2005) About the numerical solution of stationary transport equation. *WSEAS Transactions on Information Science and Applications* 9:1373-1380
10. Mokhtar-Kharroubi M (1997) *Topics in Neutron Transport Theory*. Series on Advances in Math. For Applied Sciences, Bellomo and Brezzi
11. Ntouyas SK (1997) Global existence for neutral functional integral differential equations. In: *Proc. 2nd World Congress of Nonlinear Analysts*, pp 2133-2142
12. Parton VZ, Perlin PI (1984) *Mathematical methods of the theory of elasticity*. MIR Publishers Moscow
13. Pilkuhn H (1980) *Relativistic particle physics*. Springer Verlag, New York - Heidelberg - Berlin
14. Rahnama F, Ravetto P (1998) On the equivalence on boundary and boundary condition perturbations in transport theory. *Nuclear Sci. Eng.* 128:209-222

Chapter 9

Development of genetic algorithms and C-method for optimizing a scattering by rough surface

N. Lassouaoui, H. Hafdallah Ouslimani, A. Priou

University Paris X, Nanterre, Pôle Scientifique et Technique de Ville d'Avray, Groupe Electromagnétisme Appliqué, 50 rue de Sèvre 92410, Ville d'Avray, France.

Abstract. We present in this paper the design of a frequency selective surface (FSS) with developed curvilinear coordinate based method (C-method) and genetic algorithms (GAs). The C-method analyses the rough surface and transforms the boundary-value problem to scalar Eigen equation that is solved in the spectral domain. In addition, the study of the reflectivity of a stack constituted by this rough surface and dielectric layers necessitate the development of S-matrix algorithm and shooting method. Because the GAs are powerful to explore the research spaces, in the hope to minimize the reflectivity over the studied frequency band, the synthesis with GAs is used and allows to determinate the appropriate geometry characteristics and material properties. We present numerical results in the X band frequency.

Keywords. C-Method, genetic algorithms, reflectivity, geometry characteristics, material properties, X-frequency band

9.1 Introduction

Analysis of wave scattering from periodic structures is important for various applications in the fields of electromagnetic, optic and acoustic. Rigorous methods have been developed and used [1, 3, 5, 7]. The curvilinear

coordinate method (C-Method) is now well established for analyzing the periodic surfaces; it is known to provide both simplicity and versatility [1, 5, 7]. It is used to transform the complicated boundary-value problems to the simpler, with the hope that the basic equation of the problem will remain solvable in the new space. The TE and TM fields are derived from the same scalar Eigen equation in which the geometry appears with derivative of the profile function.

Genetic algorithms GAs have become very popular for the solutions of electromagnetic design problems, here in; we propose to use them for finding the geometry characteristics and material properties of the frequency selective surface with ensuring the minimization of the reflectivity over the X-frequency band.

This paper will proceed as follows: In Sect. 9.2, we present the analysis of the frequency selective surface and the compute of the reflection. By Sect. 9.3, we give the synthesis with GAs. Then, we present the numerical results of the study. Finally, we give the conclusions and perspectives.

9.2 Compute the reflectivity

We consider a structure with a media invariant along the z axis in the Cartesian coordinate system ($Oxyz$), periodic along x axis with period d_x . The surface (S) is described by an equation: $y = a(x)$. We suppose that this structure is illuminated by a plane wave u_i propagated in the vacuum with an incidence angle θ_i from the normal and k_i is wave number.

The considered diffraction problem is to find the scattered field which satisfies [3, 5, 6, 7]:

- Helmholtz equation in each media;
- Quasi-periodicity of the field U , where U is E_z in TE mode or H_z in the TM mode;
- The boundary condition (N is the normal vector on the surface):

$$U(x, a(x)) = 0 \text{ in TE mode and } \left. \frac{dU(x, y)}{dN} \right|_{y=a(x)} = 0 \text{ in TM mode} \quad (9.1)$$

- Radiation condition is that outside certain vicinity of contour L , the scattered field must be presented with Expansion of Rayleigh [5, 7].

It is necessary to simplify Eqs. (9.1) without disturbing other conditions. For that, we seek a coordinate system which coincides with the surface S . The C-method is simplest one and translates the coordinate system by:

$$x = u, y = v + a(x), z = w \quad (9.2)$$

The boundary conditions become simple, since $v = 0$ is the profile surface where the tangential components of the fields are continuous.

The Maxwell' equations in the new coordinate system have been derived [1, 5], and the propagation equation in both regions $v > 0$ and $v < 0$ is:

$$\begin{pmatrix} k_j^2 + \frac{\partial^2}{\partial u^2} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} U_j \\ R_j \end{pmatrix} = \begin{pmatrix} i \left(\frac{\partial}{\partial u} a' + a' \frac{\partial}{\partial u} \right) & 1 + a'^2 \\ 1 & 0 \end{pmatrix} \times \frac{1}{i} \frac{\partial}{\partial v} \begin{pmatrix} U_j \\ R_j \end{pmatrix} \quad (9.3)$$

where $j=1, 2$ ($j=1$ corresponds to periodic media and $j=2$ to vacuum).

The solution $\begin{pmatrix} U_j \\ R_j \end{pmatrix}$ can be searched with an $\exp(i\rho v)$ dependence, where ρ will be a diagonal matrix whose elements are the Eigen values of the matrix M , which is deduced by Eq. (9.3) with $d\left(\begin{bmatrix} U_j \\ R_j \end{bmatrix}\right)/dv = M\begin{bmatrix} U_j \\ R_j \end{bmatrix}$, $[U_j]$ and $[R_j]$ contain the pseudo-Fourier coefficients of U_j and R_j .

By Eq. (9.3), we note that Eigen value problem is the same for both TE and TM polarizations, which represents one of the main interests of the method; only the matching of the solution at the interface $v = 0$ will differ. However, initial values at $v=0$ are unknown. So, we used the shooting method [5] which is used to solve the boundary-value differential problem and allows deducing the transmission matrix T_g of the grating [5].

Here in, we are interesting by the stack composed by a periodic media (upper media) and a dielectric media (on the bottom) backed by a perfect electrical conductor (Figure 9.1). The transmission matrix of a homogeneous dielectric layer T_h is well known [6]. With S -matrix propagation algorithm [5] and the various matrices T_g and T_h , we deduce the reflectivity of stack.

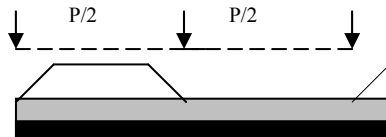


Fig. 9.1. Stack with metallic periodic media and homogeneous dielectric layer backed by a perfect electrical conductor

Knowing how we compute the reflectivity, in what follows, we present how we do the synthesis of this structure with GAs.

9.3 Synthesis with genetic algorithms

GA is started with a set of solutions (population). They work by iteratively applying genetic operators to the population to form a new one. This is motivated by a hope that the new population will be better than the old one. The solutions are selected according to the fitness; the more suitable have the more chances to reproduce. The genetic process is repeated until some condition is satisfied. Our genetic procedures are:

1. Initial population is generated randomly. To delimit the research space, we define the thresholds for each variable. Also, the theory of Maxwell equations imposes energy constraints, indeed, imaginary parts of the dielectric and magnetic constants must be positive [6, 7].
2. Coding step contains information about the problem. For the periodic media, we use a copper of a thickness of $100 \mu m$, the period P is unknown. We take the same width P_x for the two triangles and rectangle in metallic part, then, $P=6P_x$ (Figure 9.1). The dielectric layer is defined by its thickness T , the real and imaginary parts of the dielectric and magnetic constants ($\varepsilon_j = \varepsilon'_j + i\varepsilon''_j$ and $\mu_j = \mu'_j + i\mu''_j$). So, we represent the candidates by real arrays of size 6, since there are 6 variables.
3. Evaluation associates to each chromosome its reflection in dB which defines its adaptation to the problem and guides to the optimum. To optimize over X-band, we compute the reflectivity R_i for various frequencies, and the maximal R_i (which is the worst) is taken as the fitness, the goal is to ensure the minimization along all X-band.
4. Selection by population decimation: A half of the population with the candidates with fitness lowers than average are eliminating.
5. Genetic operators: With crossover and mutation, new candidates are created from the survival to replace those eliminated. There are no criteria for choosing between the crossover operators, then, from generation to generation, with equal probability, we select one operator between (mitosis, mitosis with chain reversal, N-point crossover and channeled crossing over) [2, 4]. The mutation is used to maintain the genetic diversity. If a gene is selected for mutation, then its numerical value is randomly changed by adding (or multiplying with) a random value χ . To decrease mutation effect according the generations [2,

- 4], we propose a rate Pm which decreases exponentially according the generations.
6. Stopping criteria: We chose either a maximum number of iterations; or the best fitness value is less than a threshold; or no improvement in the best fitness value for a number of generations.

9.4 Application

We present the results of optimization with GAs of the stack of Figure 9.1 for incidence normal. The research space is defined by: $(\varepsilon'_j, \mu'_j) \in]-4, 4[$, $(\varepsilon''_j, \mu''_j) \in [0, 4[$, $T \in [0.6, 10]mm$ and $P \in [25, 35]mm$.

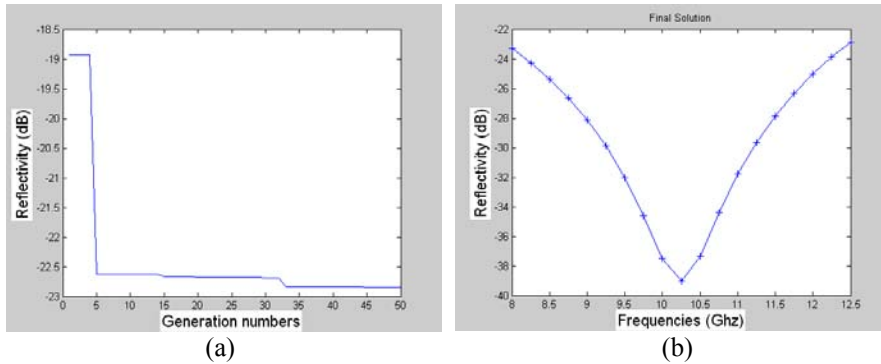


Fig. 9.2. (a) Evolution of the fitness according to the generations, (b) obtained solution at generation 50

Figure 9.2(a) shows the decrease of fitness according to the generations. Figure 9.2(b) shows the reflectivity in X-band of the final solution. After 50 generations, we reach a reflectivity of -39 dB.

For the homogeneous layer, the obtained characteristics are: $\varepsilon = -2.900 + i2.3151$, $\mu = 3.400 + i0.8805$ with the thickness of 5.3408 mm, it can be realized by a metamaterial structure. The obtained period of the rough surface is 27.5433 mm.

9.5 Conclusions

In this paper, a curvilinear coordinate based method (C-method) and genetic algorithms (GAs) are developed for designing a frequency selective

surface with the hope the minimization of the reflectivity along the X-frequency band.

The reflectivity is used as fitness function to direct the research for deducing the geometry characteristics and material properties which minimize the reflectivity over X frequency band. We chose the GAs because they are powerful for problems with a large number of dimensions and in which the solution space is unknown. Good results are obtained.

In future works, the obtained FSS structure will be simulated with 3D electromagnetic Ansoft simulator and realized using metamaterial layer for the homogeneous layer.

References

1. Granet G, Edee K, Felbacq D (2002) Scattering of a plane wave by rough surfaces: A new curvilinear coordinate system based approach. In: Progress in Electromagnetics Research PIER 37:235–250
2. Goldberg DE (1989) Genetic Algorithms in search, optimization and machine learning. Addison-Wesley
3. Kong JA (2000) Electromagnetic wave theory. EMW Publishing
4. Mitchell M (1996) Introduction to Genetic Algorithms. MIT press
5. Neviere M, Popov E (2003) Light propagation in Periodic Media, Differential theory and design. Marcel Dekker
6. Petit R (1992) Electromagnetic waves in radio electricity and optics. Masson
7. Poyedinchuk AY, Tuchkin YA, Yashina NP (2006) C-Method: Several aspects of spectral theory of gratings. In: Progress in Electromagnetics Research PIER 59:113–149

Author index

- A. Abdollahzadeh, 1427
A. El Khafaji, 563
A. G. Ravelo-García, 141
A. Garcés, R. Quirós, 355
A. Gokhan Yavuz, 663, 675, 1619
A. Jraifi, 563
A. Priou, 1639
A.P. Stasenko, 1437
A.R. Awad, 941
Abdalla Mahmoud, 1407
Adrian Curaj, 1337
Adrian Zafiu, 313, 1059, 1231
Adrian-Ioan Niculescu, 199
Afrodita Balasa Boldea, 247
Ágnes Bogárdi-Mészöly, 1241
Ahmed Sameh, 513, 1407
Al. Țugui, 1395
Alberto Salguero, 1357
Aldo Cumani, 1385
Alex Soiguine, 991
Alexandru Epureanu, 15, 1365
Ali Abbasi, 1093
Ali Olfat, 651
Ali-Reza Bagheri, 1079
Alireza Kargar, 307, 1419
Amin Khansefid, 651
Ana Maria A. C. Rocha, 1035
Ana Maria Mitu, 1319
Anca Iordan, 495
András Rövid, 163
Andreas Pomportsis, 573
Andreas Veglis, 573
Angelos Nakulas, 527
Angelos Nakulas, 537
Annely-Mihaela Zafiu, 313
Antonio Guiducci, 1385
Antonios S. Andreatos, 403
Anwar Majid Mirza, 811
Arbana Kadriu, 337
B. Punantapong, 1373
B. Satco, 1325
Bertalan Forstner, 583, 597
Bindu Goel, 59
Bruno Y. L. Kimura, 1279
C. Ciufudean, 1325
C. Filote, 1325
C. Ray Diez, 413, 433, 455
C. Rinaldi, 723
Caius Pănoiu, 495
Carlo dell'Aquila, 1173
Carlos F. Romero, 3
Carlos M. Travieso, 3
Cecilia Delgado, 1357
Celia A. Zorzo Barcelos, 191
Chao-Chin Wu, 1305, 1475, 1481, 1487, 1495, 1501
Chia-Hsun Wu, 1475
Chia-Hung Yen, 385
Chia-Jui Hsu, 1475
Chia-Jung Hsu, 1481
Chih-Hung Chung, 1481
Chi-Jen Lee, 413, 433, 443, 455
Chokri Chemak, 73
Chong-Cheng Hong, 1481
Chung-Ping Lee, 443, 1475, 1487, 1501
Chun-Hung Wu, 1475
Ciprian Cuzmin, 15, 1365
Claudius Dan, 1533
Constantin Lupsoiu, 1209
Constantin Udriște, 1597
Cornel Secara, 1319
Costin-Radu Boldea, 247
Cristea Boboila, 1209

- Csaba Török, 1469
D. C. Vasiliadis, 625, 689
Dan Dumitriu, 1319, 1543
Daniel Baldovin, 1319
Daniel Dumitriu, 199
David Anderson, 1143, 1567
David Wolfe Corne, 845
Demétrio Renó Magalhães, 43
Dongfeng Yuan, 1003
Douglas Farias Cordeiro, 191
Drago Zagar, 643
Drazen Tomic, 643
E. Camahort, 355
E. Glavas, 625
E. Stergiou, 625, 689
E.H Saidi, 563
E.S. Karapidakis, 873
E.V. Kasyanova, 483
Edita Kolářová, 1469
Edite M. G. P. Fernandes, 1035
Eduard Franti, 313
Eduard Manykin, 103
Edward Stow, 1459
Elena Doicaru, 1533
Elvira Popescu, 505
Emmanouil Zoulias, 527, 537
Esra Celik, 1619
Eugene Grichuk, 103
Ezio Lefons, 1173
F. D. Lorenzo-García, 141
F. Graziosi, 723
Farhat Saleemi, 225
Farshad Rezvan, 923
Farzad Meysami-Azad, 1105
Filippo Tangorra, 1173
France Sevšek, 837
Francisco Araque, 1357
Francisco Reinaldo, 43
Frank S. Haug, 1159
Fun-Chi Hsu, 1475
G. Atsalakis, 859
G. E. Rizos, 625, 689
G. Gargano, 723
G. Karangelis, 825
G.A. Mallah, 347
Gabriela Mircea, 609
George Popov, 235
George Savii, 495
Georgios P. Fotis 527, 537
Ghuan-Hsien Mao, 1495
Goran Martinovic, 643
Guennadi Kouzaev, 617
H. Hafdallah Ouslimani, 1639
H. Irem Turkmen, 1511
Hajar Sadeghi, 113
Hamed Arabi, 1069
Hamid Reza Shahdoosti, 1023, 1051
Hamido Fujita, 1551
Hassan Charaf, 583, 1241
Hiroshi Kawaguchi, 883
Howard Lo, 1487, 1495, 1501
Hsieh-Hua Yang, 471, 739, 763, 1125
Hsueh-Yung (Robert) Chao, 131
Hua-Lin Tsai, 413, 433, 443, 455
Huma Tauseef, 225
Hung-Jen Yang, 413, 433, 443, 455, 763, 739, 471, 1125, 1475, 1481, 1495
I. Von Poser, 941
I.O. Vardiambasis, 731
Imre Kelényi, 597
Ioannis O. Vardiambasis, 1313
Ion N. Ion, 1337
Ion Stefanescu, 1231
J. El Abbadi, 563
J. Huerta, 355
J. L. Navarro-Mesa, 141
James DeArmon, 151
James N.K. Liu, 1115
Jamie D. Mills, 375
Jane You, 1115
Jason Howarth, 1459
Javier Roca, 1043
Jean Christophe Lapayre, 73
Jesús B. Alonso, 3
Jhen-Wei Huang, 1305
Jian Ying Zhang, 27
JianDa.Han, 1575
John James, 151
John Morris, 151
José Antonio Álvarez, 1043

-
- Jose Jesús Fernández, 1043
 Jui-Chen Yu, 471, 739, 763, 1125
 K. S. Mohammed, 711
 Kavi Mahesh, 1287
 Kelly Connolly, 151
 Kentaro Kawakami, 883
 Krista Rizman Žalik, 751
 Krzysztof A. Cyran, 35
 Kuan-Chou Lai, 1305
 Kuo-Hung Tseng, 423
 L. D. Genete, 1395
 L. Pomante, 723
 Lakhdar Sahbi, 323
 Lambros Ekonomou, 527, 537
 Leonarda Carnimeo, 49
 Ligia Munteanu, 1543
 Liliana Braescu, 1345
 Ljubomir Lazić, 953
 Lucian Bărbulescu, 1533
 Luige Vladareanu, 1337
 Luís P. Reis, 43
 M. Chover, 355
 M. Elif Karsligil, 1511
 M. Goodarzi, 1427
 M. Sepehri, 1427
 M. Sepehri, 1427
 M. Amac Guvensan, 663
 M.T. Aboul-Ela, 941
 Majid Ghandchi, 923
 Manuel de Jesús Nandayapa Alfaro, 121
 Manuela Pănoiu, 495
 Marcel Migdalovici, 933
 Marcos Aurélio Batista, 191
 Margarita Kuzmina, 103
 Marian Cristian Mihăescu, 367
 Marian Cucu, 907
 Masahiko Yoshimoto, 883
 Matevž Dolenc, 1297
 MD Nor Ridzuan Daud, 845
 Mei-Huang Huang, 385
 Miguel A. Ferrer, 3
 Miguel Angel Lagunas, 1003
 Mihaela Muntean, 609
 Ming-Chang Wu, 423
 Ming-Chih Wu, 1487
 Ming-Jun Chan, 1481
 Mingyan Jiang, 1003
 Mircea Preda, 505
 Mirela-Catrinel Voicu, 893
 Mohamed Bouhdadi, 275, 287, 979
 Mohamed Salim Bouhlel, 73
 Mohammad Hossein Sheikh, 307
 Monica Dascălu, 313
 Muhammad Abuzar Fahiem, 225, 253
 N. E. Mastorakis, 87, 953, 1003, 1069, 1079, 1093, 1105, 1629
 N. Lassouaoui, 1639
 N. Tzioumakis, 731
 N.A. Shaikh, 347
 Nicolae Oancea, 1365
 Nicolae Oancea, 907
 Nisar Ahmed Memon, 811
 Nitin, 1451
 Noura Boudiaf, 323
 Olga Martin, 1011, 1629
 Omid Khayat, 1023, 1051
 Osamu Arai, 1551
 Osslan Osiris Vergara Villegas, 121
 P. Remlein, 633
 P. Tosaranon, 775
 P. Woraratsoontorn, 793, 803
 Paula Mahoney, 151
 Paulo R. M. Cereda, 1279
 Payman Moallem, 113
 Péter Várlaki, 163
 Philippe Trigano, 505
 Pofen Wang, 443
 R. Alesii, 723
 R. Orzechowski, 633
 R. Serban, 1325
 Radu Munteanu, 1337
 Rahim Ghayour, 1419
 Reginaldo A. Gotardo, 1279
 Ricardo A. Rios, 1279
 Robert Klinc, 1297
 Robson E. Grande, 1279
 Rong-Jyue Fang, 413, 433, 443, 455, 1487, 1501
 Rui Camacho, 43
 S. Albert Rabara, 1217

- S. Amirhassan Monadjemi, 113
S. Bunluechokchai, 775, 785
S. Zimeras, 825
S.A.M. Gilani, 811
Sabina-Cristiana Mihalache, 1185
Saeed K. Rahimi, 1159
Saeed-Reza Sabbagh-Yazdi, 1069,
1079, 1093, 1105
Sameer. S. Saigankar, 1255
See-Chien Hou, 423
Serena Pastore, 1269
Sérgio D. Zorzo, 1279
Shaiq A. Haq, 225
Sherif El-Kassas, 1407
Shi-Jer Lou, 385
Shivanand B. Hiremath, 1255
Simon C.K. Shiu, 1115
Simona Boboila, 1209
Sokratis K. Katsikas, 177
Sonja Petrovic-Lazarevic, 27
Stavroula Kourtesi, 527, 537
Stefan Balint, 1345
Stefan Holban, 1059
Ştefania Donescu, 1543
Stylianios Sp. Pappas, 177
Sunil Mallya, 1287
T. Leeudomwong, 793, 803
T. Melesanaki, 731
Takeshi Hashimoto, 163, 1241
Tatsuo Suzuki, 209
Thomas F. George, 1345
Tien-Sheng Tsai, 413, 433, 455
Tihamér Levendovszky, 1241
Tomáš Rebok, 265
Tomasz Śliwiński, 549
Tudor Sireteanu, 199
Tzai-Hung Huang, 385
Ulrich Albrecht, 1611
V.N. Kasyanov, 483, 1437
Valeri Mladenov, 235
Valuchandhar Vadivelu, 1217
Vasile Daniel Păvăloaia, 1197
Vasile Marinescu, 1365
Vassilios C. Moussas, 177
Veturia Chiroiu, 1319, 1543
Vianey Guadalupe Cruz Sánchez,
121
Virgil Teodor, 15, 907, 1365
Vivek Kumar Sehgal, 1451
W. Ussawongaraya, 785
Walter Dosch, 699
Wen-Hui Han, 739
Włodzimierz Ogryczak, 549
X. D. Zhuang, 87
Yasumichi Hasegawa, 209
Yi-Hui Lui, 385
Yiming Li, 131
Yi-Ting Kuo, 131
YiWen.Zhao, 1575
YiYong.Nie, 1575
Yogesh Singh, 59
Yoshinori Sakata, 883
Youssef Balouki, 275, 287, 979
Youssef Rezvan, 923
Yu-Chen Weng, 1501
Z. Cihan Taysi, 663, 675, 1619
Z.A. Shaikh, 347
Zeljko Panian, 1519
Zeyneb Kurt, 1511
Zhe.Jiang, 1575
Zoheir Kordrostami, 307
Zygmunt Ciota, 299

Index

A

Acceptance, 739
Action model, 1305
Active filters, 1533
Ad hoc networks, 690
Adaptive education systems, 483
Adaptive step length, 1003
Affine projection algorithm, 651
Agent classification, 347
Agent-oriented software engineering, 355
AI, 1143
Air traffic management, 151
Airspace congestion, 151
Alternative publishing channels, 573
Analytical calculation, 933
Analytical Hierarchy Process (AHP), 1488
Antenna arrays, 731
Anti-redressing effect, 200
Anti-roll effect, 200
Aortic stenosis, 803
Aortic stenosis, 775
ARMA, 177
Array antenna, 651
Artificial Fish Swarm Algorithm, 1003
Artificial neural network, 43, 59
Artificial pivot induction, 1287
ARTP protocol, 265
Assembly language, 1451
Assembly process, 1325
Association rule, 1115
ATM, 35
Attack, 1619
Attitude, 763

Attractors, 893
Audio surveillance, 723
Audit, 1395
Auditor, 1395
Authenticated routing for Ad Hoc Networks, 1407
Automated reasoning, 1567
Auxetic material, 1543

B

Ballistic transport, 1419
Base station positioning, 663
Beamforming, 731
Biologically inspired methods, 103
Biometric systems, 3
BitTorrent, 597
Blind adaptive multiuser detection, 651
BLM, 35
B-Rep, 253
Browsing, 1502
Brute force, 1619
Bundle adjustment, 1385

C

C#, 1469
Camera perspective, 225, 253
Carbon nanotube, 307
Case-based reasoning (CBR), 1115
Cellular automata, 247
CFA, 1125
Chaos, 893
Chess machine, 1481
ChipScope, 711
Chromosomal instability, 35
Civil engineering, 1297

Classification, 845
Clustering, 1502
Clustering, classification, 1023, 1051
C-Method, 1639
Coaxial NanowireFET, 1419
Coding of operations, 933
Color stereo matching, 113
Command module, 313
Communication history, 699
Communication technology, 527
Communications, 617
Compiler, 1437
Computational interfaces, 979
Computational objects, 979
Computer aided diagnosis, 811
Computer chess, 1481
Computer network, 643
Computer simulation, 991
Computer speech, 299
Computer system, 235
Computer-based decision models, 1185
Computerized solution, 1197
Concurrency, 1217
Concurrent, 1619
Concurrent transactions, 1209
Constitutive equation, 1011
Continuous wavelet transform, 785, 793, 803
Continuous wavelet transform, 775
Contradiction-handling, 1143
Control, 1395
Control module, 313
Convex shape, 163
Cosserat elasticity, 1543
Creative thinking abilities, 413
Cross country culture exchange project, 434
Cross media publishing, 573
CSG, 253
CT medical data, 826
current-voltage characteristics, 307

D

Damage, 1319
Data clustering, 751
Data cube, 1475
Data mining, 367
Data mining, 845, 1115
Data traffic, 367
Data warehouse, 1173, 1357, 1475
Database management systems, 1209
Deadlocks, 1209
Decision model, 1185
Deformation, 1373
Denotational Semantics, 287, 979
Dependability, 235
Dependent tasks, 625
Depth estimation, 225
Derived agentive nouns, 337
Description, 209
Design, 209, 375
Determining coefficients, 1575
Deterministic and nondeterministic networks, 1059, 1231
Diagnosis, 1325
Dictionary, 1619
Dielectric load, 1313
Diffusing vector field, 87
digital skills, 403
DIMA Model, 323
Dimensional control, 1365
Dimensional dynamics, 1365
DiProNN, 265
DiProNN program, 265
DiProNN programming model, 265
Direct sequence code division multiple access, 651
Discrete Maxwell geometric dynamics, 1597
Discrete wavelet transform, 121
Dispatching rules, 625
Distance education, 375, 385
Distance learning, 483
Distributed database, 1217
Distributed Database Systems (DDBMS), 1159

Distributed debugging, 1459
 Distributed query processing, 1159
 Distributed technologies, 1269
 Diversity, 235
 DNA computing, 751
 Document Editor, 923
 Dublin Core, 513
 DXF, 253
 Dynamical systems, 617
 Dynamic load balancing, 1043
 Dynamic plasticity, 1011
 Dynamic security assessment, 873
 Dynamic voltage scaling, 883

E

E-assessment, 403
 E-business, 1395
 Economic and financial analysis, 1197
 Economic model, 1305
 Edge-defined film-fed growth, 1345
 Edition, 323
 Educational soft-ware, 495
 Effectiveness of platform, 423
 Egomotion, 1385
 E-health service, 739
 E-health service system, 763
 Elastic pipeline, 883
 E-Learning, 505
 Electronic documents, 403
 Electronic exams, 403
 E-mail, 537
 Embedded 8-bit processor, 1451
 Embedded sram, 883
 Embedded system, 235
 Employee, 739, 763
 Energy management system, 873
 Enterprise Language, 275, 287
 Epipolar line, 113
 Equivalent transformations, 1437
 Erroneous behaviour, 699
 Error correcting codes, 73
 Errors, 1011
 Euclidean distance, 1023
 Euler scheme, 1469

Evaluation, 1059, 1231
 Evolving policies, 27
 E-Xaminer, 403
 Exsys Corvid, 1197
 EXtensible Markup Language (XML), 1159

F

Fail-safe, 235
 Fairness, 549
 Farsi language, 923
 Farsi Spell checking, 923
 FarsiTeX, 923
 Fault proneness, 59
 Fault-tolerance, 235
 Feature chain, 113
 Feature extraction, 113, 141
 Feature preserving, 121
 Feedback stabilization, 1575
 Fiber, 1345
 Field effect transistor, 307
 File Sharing, 597
 Financial Crisis Prediction, 1495
 Financial diagnose, 1197
 Fine-tune, 43
 Finite element method, 1011
 Finite volume method, 1069, 1093
 Finite-differences method, 1011
 Flexibility, 1427
 Flexible substrate, 1373
 Formal asynchronous discussion, 375
 Formal specification, 323
 Fourier coefficients, 837
 FPGA, 711
 Fractional right ideal, 1611
 Fragments, 513
 Frequency analysis, 299
 Frobenius-like form, 1575
 Functional programming, 1437
 Fundamental frequency, 299
 Fuzzy casual networks, 27
 Fuzzy control, 27
 Fuzzy techniques, 49

G

Gate oxide thickness, 1419
Generation, 209
Generative method, 1059, 1231
Genetic algorithms, 15, 731, 859, 1003, 1639
Geometry characteristics, 1639
GET, 1255
Global Data Dictionary (GDD), 1159
Global optimization, 1035
Goldie-Ring, 1611
Graphical interface, 323
Graphics 3D, 313
Grid, 1305
Grid and distributed systems, 347
Grid computing, 1297
Grid systems, 1269

H

H.264/AVC, 883
Hard tissue repair, 1373
Health Bureau, 763
Health service system, 739
Heart rate variability, 837
Heart sound analysis, 785, 793, 803
Heart sound analysis, 775
Heat function, 1023, 1051
Heat Generation and Transfer, 1079
Helical surface, 495
Hessenberg form, 1575
Heuristic algorithms, 625
Hidden Markov models, 141
High-speed circuit, 131
High-throughput computing, 1297
HMAC, 1255
Homogenisation technique, 1543
Hopf bifurcation, 609
HSPICE, 131
HTTP, 1255
HTTP_REFERER, 1255
Human body shadowing, 563
Human familial cancer, 35
Hybrid modes, 1313
Hybrid/online education, 375

Hyper plane, 1051

Hypermedia, 505

I

Ibragimov-Maxwell Lagrangian, 1597
ICT, 403
Image coding, 121
Image processing, 49, 103
Image segmentation, 87, 1003
Image structure, 87
Image transform, 87
Imaging, 837
Implementation methods, 893
Implicants, 1059, 1231
Indoor propagation, 563
Inference engines, 1185
Information retrieval, 1185, 1502
Information system, 1357
Information technology, 471, 527, 1519
Information technology, 1395
Information theory, 527
Input/output relation, 209
Inset dielectric guide, 1313
InteliGrid, 1297
Intelligent shock absorber, 200
Intelligent systems in economy, 1197
Intelligent technique, 941
Interface, 1319
Internet model, 609
Iris data, 1051

J

Java Server Pages, 1255
Java Virtual Machine, 1459
Java Virtual Machine Debugging Interface, 1459
Java web service applications, 1269
JavaSpaces, 1459
Jini, 1459
Join selectivity, 1173

K

Kalman, 177
 K-Means, 663
 K-Means Local+ Algorithm, 663
 Knowledge application, 423
 Knowledge management, 423
 Knowledge reuse, 471, 1125

L

Landauer formula, 307
 Laplace equation, 307
 Large-scale example, 941
 Laser triangulation, 225
 Learner modeling, 505
 Learning Style, 505
 Lectromagnetism-like algorithm, 1035
 Lexicographic max-min, 549
 Lexicographic optimization, 549
 Line drawings, 253
 Linear arrays, 731
 Linear discriminant analysis, 1511
 Liquid crystal display, 1451
 List scheduling algorithms, 625
 LMML, 513
 Locking, 1209
 Logic, 1143
 Logic, 1567
 Logic circuit, 1495
 Logistic regression, 59
 LOM, 513
 Lorenz plot, 837
 Low noise, 1533
 Low sensitivities, 1533
 Lyee method, 1551

M

Machine-deduction, 1143
 Maintainability, 1427
 Malicious nodes, 1407
 Manifolds, 617
 Mass transfer, 1345
 Material properties, 1639
 Mathematical modeling, 367

Mathematical morphology, 811
 Matrix of polynomials, 933
 Maude, 323
 Maxwell PDEs, 1597
 Mean-Value Analysis, 1241
 Measurement, 163
 Mechanical modeling, 933
 Medical diagnosis, 845
 Medical image analysis, 811
 Medical image segmentation, 811
 Memory bandwidth, 883
 Mesoscopic material, 1319
 Message Digest 5, 1255
 Meta-cognition, 1125
 Metadata, 513
 Metamodel, 1185
 Method of finite differences, 1629
 Metrics, 59
 Micropolar theory, 1319
 Microslot lines, 1313
 Milstein scheme, 1469
 MIMO channel, 633
 Mission Planning, 1395
 Mitral regurgitation, 785, 793
 Mobile ad hoc networks, 1407
 Mobile computing, 513
 Mobile networks, 690
 Mobile Networks, 597
 Mobile project, 443
 Mobile Technology, 456
 Model-view-controller, 1255
 Morphological segmentation, 337
 Multiagent systems, 347
 Multi-agents systems, 323
 Multi-dimensional database, 1475
 Multi-dimensional signals, 617
 Multi-layer algorithm, 1069
 Multilevel security, 1217
 Multi-objective optimization, 731
 Multiprocessor systems, 1337
 Multi-resolution field, 73
 Multi-threshold, 1003
 Multi-valued combinational minimization, 1059, 1231
 Multivariate, 177
 Murmur, 775, 785, 793, 803

N

NASIR Finite Volume, 1079, 1105
Natural selection, 35
Negative Poisson's ratio, 1543
NET platform, 347
Network, 609
Network and parallel minimization, 1059, 1231
Network design, 549
Networked embedded systems, 723
Networks of coupled oscillators, 103
Neural networks, 3, 15, 859, 1495
Neural networks models, 27
Neutrality tests, 35
Neutron transport equation, 1629
Non binary turbo encoding, 633
Nonlinear systems, 893
Non-singular ring, 1611
Numerical methods, 991
Numerical simulation, 609
Numerical stabilization, 1575

O

Objective function, 1325
Object-oriented languages, 991
Object-oriented methodology, 1427
OLAP, 1475
Online identification, 1365
On-line web learning, 434
Ontology, 1185
Open-architecture systems, 1337
Operational analysis, 643
Optimization, 941, 1437
Orff music teaching method, 413
Organizational innovativeness, 763
Orthographic projections, 253
Ottoman character recognition, 1511
OWL, 1185
OWL-S, 1269

P

Paraconsistency, 1143, 1567
Parallel algorithms, 991

Parallel iterative reconstruction algorithms, 1043
Parallel programming, 1437
Parameterization, 3
Parametric studies, 1297
Particle swarm optimization, 1003
Particle systems, 191
Partition ratio, 1287
Partitioning, 177
Password recovery, 1619
Pattern, 209
Pattern marching, 1481
Pattern search method, 1035
Peak-analysis, 1287
Pedagogy, 495
Peer-to-Peer Protocols, 597
Performance, 1217
Performance monitoring, 643
Performance prediction, 1241
Personal data protection, 1279
Personal innovativeness, 763
Personalization, 1279
Petri nets, 151, 1325
Phantoms, 1209
Phonocardiography, 775, 785, 793, 803
PHS phone, 443
Positional value criterion, 625
POST, 1255
Potential distribution, 1419
Power systems, 873
Power-aware sensors, 675
PPM, 711
Preparing engineering high school teachers, 423
Principal component analysis, 1511
Privacy, 1279
Probabilistic neural networks, 35
Processing redundancy, 1059, 1231
Programming, testing, 483
Project- based learning (PBL), 1488
Projection selectivity, 1173
Proportional share, 1305
Prosthesis, 313
Protégé, 1185

Q

Query cost estimation, 1173
 Query optimization, 1173
 Queueing model, 1241

R

Radio propagation channel, 563
 Rare earth compounds, 1345
 RCC Dam Construction Program, 1079
 RDF, 513, 1185
 Reaction-diffusion systems, 247
 Reactive program, 1551
 Real genetic algorithms, 941
 Real time control, 1337
 Realization, 209
 Real-time applications, 163
 Reasoning, 1143
 Reconfigurable machine tools, 1365
 Reconstruction 3D, 253
 RECQL, 35
 Reflectivity, 1639
 Reliability, 1427
 Rendering techniques, 826
 Reputation-based Schemes, 1407
 Requirements engineering, 1519
 Requirements management, 1519
 Resource allocation, 549
 Revolution surface, 495
 Rewriting Logic, 323
 Risk, 1395
 RM-ODP, 275, 287, 979
 Robot localisation, 1385
 Robustness, 73
 ROC curve, 59
 Routing protocols, 690
 Routing Security, 1407
 RPSNR, 73
 RS codes, 633
 Rule induction algorithm, 845

S

scanners 3D, 225
 Scanning 3D, 163

Scheduling problems, 625
 Schottky Barrier, 307, 1419
 Science and technology curriculum, 456
 Searching space, 1481
 Second order Godunov Type
 Reiman Solver, 1093
 segmentation techniques, 826
 Self organizing map, 1502
 Selfish nodes, 1407
 SEM, 471
 Semantic tableaux, 1143, 1567
 Semantic technologies, 1269
 Semantic web services, 1269
 Semantics, 275
 Semi cylinders, 1313
 Senior-vocational-high, 471
 Sensor node design, 675
 Sequent calculus, 1567
 Sequential memory component, 699
 Series uniformly convergent, 1629
 Service discovery, 1269
 Service domain, 699
 SGS Turbulent Viscosity Model, 1105
 Shallow water equations, 1069
 Shape reconstruction, 826
 Shells, 933
 Sidelobe level, 731
 Simulation, 191, 313, 323, 690
 Simulation model, 151
 Simultaneous localisation and mapping, 1385
 Single crystal growth, 1345
 Singular equations, 1313
 Skyhook, 200
 Sleep quality, 141
 Smart antennas, 731
 Software agents, 355
 Software engineering, 1519
 Software methodologies, 355
 Software test effort, 954
 Software test points, 954
 Software/System testing, 954
 Space time encoding, 633
 Spam, 537

SPARQL, 1185
Sparsely-connected Neural
 Networks, 49
Speech processing, 299
SPIHT, 121
Spit, 537
State estimation function, 1325
State transition machine, 699
State-space equation, 131
Statistics learning, 375
Stepper motor, 1451
Stereo vision, 113, 163, 1385
Stochastic differential equation,
 1469
Storage tanks interference, 1105
Stratified Sampling, 43
Stream function, 699
Stream processing, 265
String sorting, 1287
Strong unstable predicates, 1459
Structural Concepts, 287
Structured light, 225
Structured Query Language (SQL),
 1159
Structured System Analysis and
 Design Methodology, 1427
SURF, 1385
Surface identification, 15
Surfaces discreet representation, 907
SVG, 253
Symbolic computation, 495
Synchronization, 103
Synthesis, 1533
System integration, 1357

T

Tangents method, 907
Teaching and learning, 375
Team teaching, 385
Technology integrate into
 instruction, 456
Temporal logic, 1551
Test management, 954
Test metrics, 954
Theorem-proving, 1143

Thread pool, 1241
Topological computing, 617
Topological schemes, 617
Topology, 617
Transaction, 1217
Trapezoidal rule, 1629
Traveling salesman problem, 941
Triangle Chess, 1481
Triangle Unstructured Mesh, 1105
Tropical cyclone forecast, 1115
Tunneling probability, 307
Turbo code, 73
Two-level morphology, 337

U

UDDI, 1269
Udriște-Maxwell Lagrangian, 1597
Ultra-discretisation, 247
UML/OCL, 275, 287, 979
UMPS, 1451
Unsolicited messages, 537
Unsteady pipe flow, 1093
URL, 1255
Usability, 1427
User level threads, 1043
User requirement, 1551
User requirements, 1519
Utilization, 643
UWB, 711

V

Vector clock, 1459
Vector space model, 1502
Verification, 1551
Vertical acceleration, 200
VHDL, 711
Virtual environments, 313
Viscoplasticity, 1319
Visualization, 191
Vocal tract, 299
VoIP, 537
Volume definition, 826
Von Neumann analysis, 1597

W

Walking robots, 1337
Water-Hammer, 1093
Watermarking, 73
Wave propagation, 1313
Wavelet based contourlet transform, 121
Wavelet entropy, 775, 793
Weak unstable predicates, 1459
Web, 1279
Web application, 1241, 1255
Web portal, 1297
Web privacy preferences, 1279
Web repositories, 513
Web-based education, 483
Wind energy, 859
Wind flow, 1105
Wind induced flow, 1069
Wind power forecasting, 859
Wireless networks, 690
Wireless sensor networks, 663, 723

Wireless sensor node, 675
WKB approximation, 307
Workflow management, 573
Worm cutter profiling, 907
Wrapper, 43
Writer identification, 3
WRN, 35
WSDL, 1269
WSN, 675

X

X-frequency band, 1639
Xilinx, 711

Y

Young's modulus, 1543

Z

Zip, 1619