

Jasni Mohamad Zain
Wan Maseri bt Wan Mohd
Eyas El-Qawasmeh (Eds.)

Communications in Computer and Information Science

181

Software Engineering and Computer Systems

Second International Conference, ICSECS 2011
Kuantan, Pahang, Malaysia, June 2011
Proceedings, Part III

Part 3

Communications
in Computer and Information Science

181

Jasni Mohamad Zain Wan Maseri bt Wan Mohd
Eyas El-Qawasmeh (Eds.)

Software Engineering and Computer Systems

Second International Conference, ICSECS 2011
Kuantan, Pahang, Malaysia, June 27-29, 2011
Proceedings, Part III

Volume Editors

Jasni Mohamad Zain
Wan Maseri bt Wan Mohd
Universiti Malaysia Pahang
Faculty of Computer Systems and Software Engineering
Lebuhraya Tun Razak, 26300 Gambang, Kuantan, Pahang, Malaysia
E-mail: {jasni, maseri}@ump.edu.my

Eyas El-Qawasmeh
King Saud University
Information Systems Department
Riyadh 11543, Saudi Arabia
E-mail: eyasa@usa.net

ISSN 1865-0929
ISBN 978-3-642-22202-3
DOI 10.1007/978-3-642-22203-0
Springer Heidelberg Dordrecht London New York

e-ISSN 1865-0937
e-ISBN 978-3-642-22203-0

Library of Congress Control Number: 2011930423

CR Subject Classification (1998): H.4, H.3, D.2, C.2, F.1, I.4-5

© Springer-Verlag Berlin Heidelberg 2011

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Message from the Chairs

The Second International Conference on Software Engineering and Computer Systems (ICSECS 2011) was co-sponsored by Springer is organized and hosted by the Universiti Malaysia Pahang in Kuantan, Pahang, Malaysia, from June 27-29, 2011, in association with the Society of Digital Information and Wireless Communications. ICSECS 2011 was planned as a major event in the software engineering and computer systems field, and served as a forum for scientists and engineers to meet and present their latest research results, ideas, and papers in the diverse areas of data software engineering, computer science, and related topics in the area of digital information.

This scientific conference included guest lectures and 190 research papers that were presented in the technical session. This meeting was a great opportunity to exchange knowledge and experience for all the participants who joined us from all over the world to discuss new ideas in the areas of software requirements, development, testing, and other applications related to software engineering. We are grateful to the Universiti Malaysia Pahang in Kuantan, Malaysia, for hosting this conference. We use this occasion to express thanks to the Technical Committee and to all the external reviewers. We are grateful to Springer for co-sponsoring the event. Finally, we would like to thank all the participants and sponsors.

Jasni Mohamad Zain
Wan Maseri Wan Mohd
Hocine Cherifi

Preface

On behalf of the ICSECS 2011 Program Committee and the Universiti Malaysia Pahang in Kuantan, Pahang, Malaysia, we welcome readers to proceedings of the Second International Conference on Software Engineering and Computer Systems (ICSECS 2011).

ICSECS 2011 explored new advances in software engineering including software requirements, development, testing, computer systems, and digital information and data communication technologies. It brought together researchers from various areas of software engineering, information sciences, and data communications to address both theoretical and applied aspects of software engineering and computer systems. We do hope that the discussions and exchange of ideas will contribute to advancements in the technology in the near future.

The conference received 530 papers, out of which 205 were accepted, resulting in an acceptance rate of 39%. These accepted papers are authored by researchers from 34 countries covering many significant areas of digital information and data communications. Each paper was evaluated by a minimum of two reviewers.

We believe that the proceedings document the best research in the studied areas. We express our thanks to the Universiti Malaysia Pahang in Kuantan, Malaysia, Springer, the authors, and the organizers of the conference.

Jasni Mohamad Zain
Wan Maseri Wan Mohd
Hocine Cherifi

Organization

Program Co-chairs

Yoshiro Imai

Renata Wachowiak-Smolikova

Eyas El-Qawasmeh

Kagawa University, Japan

Nipissing University, Canada

King Saud University, Saudi Arabia

Publicity Chairs

Ezendu Ariwa

Jan Platos

Zuqing Zhu

London Metropolitan University, UK

VSB-Technical University of Ostrava, Czech
Republic

University of Science and Technology of
China, China

Table of Contents – Part III

Software Design/Testing

Practical Adoptions of T-Way Strategies for Interaction Testing	1
<i>Kamal Z. Zamli, Rozmie R. Othman, Mohammed I. Younis, and Mohd Hazli Mohamed Zabil</i>	
On the Relationship between Proof Writing and Programming: Some Conclusions for Teaching Future Software Developers	15
<i>Michael Hartwig</i>	
Model-Based Web Components Testing: A Prioritization Approach	25
<i>Ahmed Al-Herz and Moataz Ahmed</i>	
Web Interactive Multimedia Technology: State of the Art	41
<i>Asma Md Ali and Joan Richardson</i>	
Health Architecture Based on SOA and Mobile Agents	54
<i>Mohamed Elammari and Tarek F. Alteap</i>	
Metrics Based Variability Assessment of Code Assets	66
<i>Fazal-e-Amin, Ahmad Kamil Bin Mahmood, and Alan Oxley</i>	
Test Data Generation for Event-B Models Using Genetic Algorithms . . .	76
<i>Ionut Dinca, Alin Stefanescu, Florentin Ipate, Raluca Lefticaru, and Cristina Tudose</i>	
A Parallel Tree Based Strategy for T-Way Combinatorial Interaction Testing	91
<i>Mohammad F.J. Klaib, Sangeetha Muthuraman, and Ahmed Noraziah</i>	
PS2Way: An Efficient Pairwise Search Approach for Test Data Generation	99
<i>Sabira Khatun, Khandakar Fazley Rabbi, Che Yahaya Yaakub, M.F.J. Klaib, and Mohammad Masroor Ahmed</i>	
The Preferable Test Documentation Using IEEE 829	109
<i>Roslina Mohd Sidek, Ahmed Noraziah, and Mohd Helmy Abd Wahab</i>	
Automatic Analysis of Static Execution Time for Complex Loop Contained External Input	119
<i>Yun-Kwan Kim, Doo-Hyun Kim, Tae-Wan Kim, and Chun-Hyon Chang</i>	

Software Reuse: MDA-Based Ontology Development to Support Data Access over Legacy Applications	130
<i>Heru-Agus Santoso, Su-Cheng Haw, and Chien-Sing Lee</i>	
A Framework to Assure the Quality of Sanity Check Process	143
<i>Rabia Sammi, Iram Masood, and Shunaila Jabeen</i>	
Scalability of Database Bulk Insertion with Multi-threading	151
<i>Boon Wee Low, Boon Yaik Ooi, and Chee Siang Wong</i>	
Towards Unit Testing of User Interface Code for Android Mobile Applications	163
<i>Ben Sadeh, Kjetil Ørbekk, Magnus M. Eide, Njaal C.A. Gjerde, Trygve A. Tønnesland, and Sundar Gopalakrishnan</i>	
Security Modeling of SOA System Using Security Intent DSL	176
<i>Muhammad Qaiser Saleem, Jafreezal Jaafar, and Mohd Fadzil Hassan</i>	
An Input-Driven Approach to Generate Class Diagram and Its Empirical Evaluation	191
<i>Faridah Hani Mohamed Salleh</i>	
Understanding Motivators and De-motivators for Software Engineers – A Case of Malaysian Software Engineering Industry	205
<i>Mobashar Rehman, Ahmad Kamil Bin Mahmood, Rohani Salleh, and Aamir Amin</i>	
UML Diagram for Design Patterns	215
<i>Muhazam Mustapha and Nik Ghazali Nik Daud</i>	
Towards Natural Interaction with Wheelchair Using Nintendo Wiimote Controller	231
<i>Mahmood Ashraf and Masitah Ghazali</i>	
Meta-model Validation of Integrated MARTE and Component-Based Methodology Component Model for Embedded Real-Time Software	246
<i>Mohd Zulkifli M. Zaki, Mohd Adham Isa, and Dayang N.A. Jawawi</i>	
Abstract Formal Framework for Method Overriding	257
<i>Siti Hafizah, Mohd Sapiyan Baba, and Abdullah Gani</i>	
Parametric Software Metric	266
<i>Won Shin, Tae-Wan Kim, Doo-Hyun Kim, and Chun-Hyon Chang</i>	
E- Technology	
Automatic Recognition of Document Structure from PDF Files	274
<i>Rosmayati Mohemad, Abdul Razak Hamdan, Zulaiha Ali Othman, and Noor Maizura Mohamad Noor</i>	

Comparative Evaluation of Semantic Web Service Composition Approaches	283
<i>Radziah Mohamad and Furkh Zeshan</i>	
Ontology Development for Programming Related Materials	291
<i>Siti Noradlina Mat Using, Rohiza Ahmad, and Shakirah Mohd. Taib</i>	
User-Centered Evaluation for IR: Ranking Annotated Document Algorithms	306
<i>Syarifah Bahiyah Rahayu, Shahrul Azman Noah, and Andrianto Arfan Wardhana</i>	
Ad Hoc Networks	
Efficient Wireless Communications Schemes for Machine to Machine Communications	313
<i>Ronny Yongho Kim</i>	
Efficient Data Transmission Scheme for Ubiquitous Healthcare Using Wireless Body Area Networks	324
<i>Cecile Kateretse and Eui-Nam Huh</i>	
An Algorithm to Detect Attacks in Mobile Ad Hoc Network	336
<i>Radhika Saini and Manju Khari</i>	
Integrated Solution Scheme with One-Time Key Diameter Message Authentication Framework for Proxy Mobile IPv6	342
<i>Md. Mahedi Hassan and Poo Kuan Hoong</i>	
A Collaborative Intrusion Detection System against DDoS Attack in Peer to Peer Network	353
<i>Leila Ranjbar and Siavash Khorsandi</i>	
Impact of Safety Beacons on the Performance of Vehicular Ad Hoc Networks	368
<i>Bilal Munir Mughal, Asif Ali Wagan, and Halabi Hasbullah</i>	
Analysis of Routing Protocols in Vehicular Ad Hoc Network Applications	384
<i>Mojtaba Asgari, Kasmiran Jumari, and Mahamod Ismail</i>	
Comparative Study on the Performance of TFRC over AODV and DSDV Routing Protocols	398
<i>Khuzairi Mohd Zaini, Adib M. Monzer Habbal, Fazli Azzali, and Mohamad Rizal Abdul Rejab</i>	
An Enhanced Route Discovery Mechanism for AODV Routing Protocol	408
<i>Kamarularifin Abd. Jalil, Zaid Ahmad, and Jamalul-Lail Ab Manan</i>	

Fast Handover Technique for Efficient IPv6 Mobility Support in Heterogeneous Networks 419
Radhwan M. Abdullallah, Nor Asilah Wati Abdul Hamid, Shamala K. Subramaniam, and Azizol Abdullah

Using Dendritic Cell Algorithm to Detect the Resource Consumption Attack over MANET 429
Maha Abdelhaq, Rosilah Hassan, and Raed Alsaqour

Social Networks

Modeling, Analysis, and Characterization of Dubai Financial Market as a Social Network 443
Ahmed El Toukhy, Maytham Safar, and Khaled Mahdi

Secret-Eye: A Tool to Quantify User’s Emotion and Discussion Issues through a Web-Based Forum 455
Siti Z.Z. Abidin, Nasiroh Omar, Muhammad H.M. Radzi, and Mohammad B.C. Haron

Software Process Modeling

A Concurrent Coloured Petri Nets Model for Solving Binary Search Problem on a Multicore Architecture 463
Alaa M. Al-Obaidi and Sai Peck Lee

An Approach for Source Code Classification Using Software Metrics and Fuzzy Logic to Improve Code Quality with Refactoring Techniques 478
Pornchai Lerthathairat and Nakornthip Prompoon

Balanced Hierarchical Method of Collision Detection in Virtual Environment 493
Hamzah Asyrani Sulaiman and Abdullah Bade

An Aspect Oriented Component Based Model Driven Development 502
Rachit Mohan Garg and Deepak Dahiya

Application of 80/20 Rule in Software Engineering Rapid Application Development (RAD) Model 518
Muhammad Rizwan and Muzaffar Iqbal

Development of a Framework for Applying ASYCUDA System with N-Tier Application Architecture 533
Ahmad Pahlavan Tafti, Safoura Janosepah, Nasser Modiri, Abdolrahman Mohammadi Noudeh, and Hadi Alizadeh

An Experimental Design to Exercise Negotiation in Requirements Engineering	542
<i>Sabrina Ahmad and Noor Azilah Muda</i>	
A Study of Tracing and Writing Performance of Novice Students in Introductory Programming	557
<i>Affandy, Nanna Suryana Herman, Sazilah Binti Salam, and Edi Noersasongko</i>	
A Review of Prominent Work on Agile Processes Software Process Improvement and Process Tailoring Practices	571
<i>Rehan Akbar, Mohd Fadzil Hassan, and Azrai Abdullah</i>	
An Evaluation Model for Software Reuse Processes	586
<i>Anas Bassam AL-Badareen, Mohd Hasan Selamat, Marzanah A. Jabar, Jamilah Din, and Sherzod Turaev</i>	
Achieving Effective Communication during Requirements Elicitation - A Conceptual Framework	600
<i>Fares Anwar, Rozilawati Razali, and Kamsuriah Ahmad</i>	
Investigating the Effect of Aspect-Oriented Refactoring on Software Maintainability	611
<i>Hamdi A. Al-Jamimi, Mohammad Alshayeb, and Mahmoud O. Elish</i>	
On the Modelling of Adaptive Hypermedia Systems Using Agents for Courses with the Competency Approach	624
<i>Jose Sergio Magdaleno-Palencia, Mario Garcia-Valdez, Manuel Castanon-Puga, and Luis Alfonso Gaxiola-Vega</i>	
Toward a Methodological Knowledge for Service-Oriented Development Based on OPEN Meta-Model	631
<i>Mahdi Fahmideh, Fereidoon Shams, and Pooyan Jamshidi</i>	
Conceptual Framework for Formalizing Multi-Agent Systems	644
<i>Tawfiq M. Abdelaziz</i>	
Miscellaneous Topics in Software Engineering and Computer Systems	
Mining Optimal Utility Incorporated Sequential Pattern from RFID Data Warehouse Using Genetic Algorithm	659
<i>Barjesh Kochar and Rajender Singh Chhillar</i>	
SYEDWSIM: A Web Based Simulator for Grid Workload Analysis	677
<i>Syed Nasir Mehmood Shah, Ahmad Kamil Bin Mahmood, and Alan Oxley</i>	

F-IDS: A Technique for Simplifying Evidence Collection in Network Forensics	693
<i>Eviyanti Saari and Aman Jantan</i>	
Deterministic-Rule Programs on Specialization Systems: Clause-Model Semantics	702
<i>Kiyoshi Akama, Ekawit Nantajeewarawat, and Hidekatsu Koike</i>	
A Novel Replica Replacement Strategy for Data Grid Environment	717
<i>Mohammed Madi, Yuhanis Yusof, Suhaidi Hassan, and Omar Almomani</i>	
Investigate Spectrum-Sliced WDM System for FTTH Network	728
<i>Nasim Ahmed, S.A. Aljunid, R.B. Ahmad, Hilal Adnan Fadil, and M.A. Rashid</i>	
Use Case-Based Effort Estimation Approaches: A Comparison Criteria	735
<i>Mohammed Wajahat Kamal, Moataz Ahmed, and Mohamed El-Attar</i>	
An Agent-Based Autonomous Controller for Traffic Management	755
<i>Sadia Afsar, Abdul Mateen, and Fahim Arif</i>	
Comparative Evaluation of Performance Assessment and Modeling Method For Software Architecture	764
<i>Mohd Adham Isa and Dayang N.A. Jawawi</i>	
A New Approach Based on Honeybee to Improve Intrusion Detection System Using Neural Network and Bees Algorithm	777
<i>Ghassan Ahmed Ali and Aman Jantan</i>	
Multi-classifier Scheme with Low-Level Visual Feature for Adult Image Classification	793
<i>Mohammadmehdi Bozorgi, Mohd Aizaini Maarof, and Lee Zhi Sam</i>	
Managing Communications Challenges in Requirement Elicitation	803
<i>Noraini Che Pa and Abdullah Mohd Zin</i>	
Learning Efficiency Improvement of Back Propagation Algorithm by Adaptively Changing Gain Parameter together with Momentum and Learning Rate	812
<i>Norhamreeza Abdul Hamid, Nazri Mohd Nawi, Rozaida Ghazali, and Mohd Najib Mohd Salleh</i>	
Author Index	825

Practical Adoptions of T-Way Strategies for Interaction Testing

Kamal Z. Zamli¹, Rozmie R. Othman², Mohammed I. Younis¹,
and Mohd Hazli Mohamed Zabil²

¹ School of Electrical Engineering, Universiti Sains Malaysia,
Engineering Campus, Nibong Tebal
14300 Penang, Malaysia

² School of Computer and Communication
Universiti Malaysia Perlis (UniMAP)
PO Box 77, d/a Pejabat Pos Besar
01007 Kangar, Perlis, Malaysia

eekamal@eng.usm.my, rozmie.razif.othman@gmail.com,
younismi@gmail.com, mhmz11_eee078@student.usm.my

Abstract. This paper discusses the practical adoption of t-way strategies (also termed interaction testing) for interaction testing. Unlike earlier work, this paper also highlights and unifies the different possible use of t-way strategies including uniform interaction, variable strength interaction, and input-output based relations. In order to help engineers make informed decision on the different use of t-way strategies, this paper discusses the main issues and shortcomings to be considered as well as demonstrates some practical results with a-step-by-step example. In doing so, this paper also analyzes the related works highlighting the current state-of-the-arts and capabilities of some of the existing t-way strategy implementations.

Keywords: software testing, interaction testing, t-way strategies, multi-way testing, combinatorial testing.

1 Introduction

The demand for multi-functional software has grown drastically over the years. To cater this demand, software engineers are forced to develop complex software with increasing number of input parameters. As a result, more and more dependencies between input parameters are to be expected, opening more possibilities of faults due to interactions. Although traditional static and dynamic testing strategies (e.g. boundary value analysis, cause and effect analysis and equivalent partitioning) are useful in fault detection and prevention [1], however they are not designed to detect faults due to interaction. As a result, many researchers nowadays are focusing on sampling strategy that based on interaction testing (termed t-way testing) [2].

As far as t-way testing is concerned, 3 types of interaction can be associated with interaction testing (i.e. uniform strength interaction, variable strength interaction and input-output based relations). Although one interaction type has advantages over the

others in certain cases, however, no single interaction type can claim to be the ultimate solution for all interaction testing problems. Motivated by this challenge, this paper unifies and highlights the different possible use of t-way strategies. In order to help engineers make informed decision on the different use of t-way strategies, this paper also discusses the main issues and shortcomings to be considered as well as demonstrates some practical results with a-step-by-step example. In doing so, this paper also analyzes the related works highlighting the current state-of-the-arts and capabilities of some of the existing t-way strategy implementations.

For the purpose of presentation, the rest of this paper is organized as follows. Section 2 discusses fundamental of t-way strategies. Section 3 demonstrates the running example. Section 4 highlights our observations and issues. Section 5 analyses the related works. In Section 6, we present the digest of our analysis. Finally, section 7 summarizes our conclusion.

2 Fundamental of T-Way Strategies

Mathematically, t-way strategies can be abstracted to a covering array. Throughout this paper, the symbols p , v , and t are used to refer to number of parameters (or factor), values (or levels) and interaction strength for the covering array respectively. Referring to Table 1, the parameters are A, B, C, and D whilst the values are (a1, a2, b1, b2, c1, c2).

Earlier works suggested three definitions for describing the covering array. The first definition is based on whether or not the numbers of values for each parameter are equal. If the number of values is equal (i.e. uniformly distributed), then the test suite is called Coverage Array (CA). Now, if the number of values is non-uniform, then the test suite is called Mixed Coverage Array (MCA) [3, 4]. Finally, Variable Strength Covering array (VCA) refers to case when a smaller subset of covering arrays (i.e. CA or MCA) constitutes a larger covering array.

Although useful, the aforementioned definitions do not cater for the fact that there could be consideration of input and output (IO) based relations in order to construct CA, MCA, and VCA. As will be seen later, building from VCA notation for covering array, we have introduced a workable notation for IO based relations.

Normally, the CA takes parameters of N , t , p , and v respectively (i.e. $CA(N,t,p,v)$). For example, $CA(9, 2, 4, 3)$ represents a test suite consisting of 9×4 arrays (i.e. the rows represent the size of test cases (N), and the column represents the parameter (p)). Here, the test suite also covers 2-way interaction for a system with 4 3 valued parameter.

Alternatively, MCA takes parameters of N , t , and Configuration (C) (i.e. $MCA(N,t,C)$). In this case, N and t carries the same meaning as in CA. Here, C captures the parameters and values of each configuration in the following format: $v_1^{p1} v_2^{p2}, \dots, v_n^{pn}$ indicating that there are $p1$ parameters with $v1$ values, $p2$ parameters with $v2$ values, and so on. For example, $MCA(1265, 4, 10^2 4^1 3^2 2^7)$ indicates the test size of 1265 which covers 4-way interaction. Here, the configuration takes 12 parameters: 2 10 valued parameter, 1 4 valued parameter, 2 3 valued parameter and 7 2 valued parameter. Such notation can also be applicable to CA (e.g. $CA(9,2,4,3)$ can be rewritten as $CA(9,2,3^4)$).

In the case of VCA, the parameter consists of N , t , C , and Set (S) (i.e. VCA (N,t,C,S)). Similar to MCA, N,t , and C carry the same meaning. Set S consists of a multi-set of disjoint covering array with strength larger t . For example, VCA (12, 2, $3^2 2^2$, {CA (3, $3^2 2^2$)}) indicates the test size of 12 for pairwise interaction (with 2 3 valued parameter and 2 2 valued parameter) and 3-way interaction (with 1 3 valued parameter and 2 2 valued parameter). As a special case of VCA, we can also consider cumulative combination of interaction. Using the same example, we can have VCA (14, {CA (2, $3^2 2^2$)}, {CA (3, $3^2 2^2$)}). Here, we have the test size of 14 for both pairwise and 3-way interaction (with with 2 3 valued parameter and 2 2 valued parameter).

In order to expand the scope of covering arrays for IO based relations, there is a need for a more compact notation. Here, building from CA, MCA, and VCA notation, we can express IO base relations (IOR) as IOR (N, C, R). Here, N and C take the same meaning given earlier whilst R represents a multi set of parameter relationship definition contributing towards the outputs. For example, for a 4 parameters system with 2 values and each parameter will be assigned a number 0, 1, 2 and 3 respectively. Assume two input-output relationships involve in the outputs (i.e. the first and the last parameter for the first output and the second and third parameter for the second output). Here, the relationship is written as $R = \{\{0, 3\}, \{1, 2\}\}$. Assuming the test size is 12, the complete notation for can be expressed as IOR (12, 4^2 , $\{\{0,3\}, \{1,2\}\}$).

3 Running Example

In order to aid the discussion, consider the following software system example in Fig. 1.

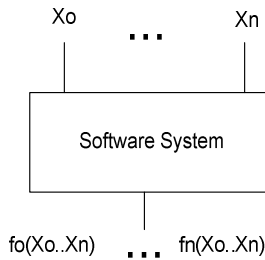


Fig. 1. Model of a Typical Software System

Assume that the input set $X = \{x_0 \dots x_n\}$ significantly affects the output, noted as $fo(x_0 \dots x_n)$ to $fn(x_0 \dots x_n)$. If X is known to take a set of data values: $D(x_0), D(x_1) \dots D(x_n)$, then the system must be tested against the set of all possible combinations of D . Here, the result is an ordered n -tuples $\{d_0, d_1 \dots d_n\}$ where each d_i is an element of $D(x_i)$. The size of the test suite would be the product size of all $D(x)$:

$$T_{\text{suite}} = \{ D(x_0) \times D(x_1) \times \dots \times D(x_n) \} \tag{1}$$

Obviously, the test suite T_{suite} can grow exponentially with the increase size of data element in the set $D(x_0), D(x_1) \dots D(x_n)$. As far as the actual test data of T_{suite} is concerned, one can consider the interaction between all n variables $x_0, x_1, x_2 \dots x_n$,

termed, *exhaustive test*. Optionally, one can also consider the interaction of any t-way interactions of variables. Here, the value of t can take the minimum of 2 and the maximum of n-1. As a running example, let us assume that the starting test case for X, termed *base test case*, has been identified in Table 1. Here, symbolic values (e.g. a1, a2, b1, b2, c1, c2) are used in place of real data values to facilitate discussion.

Here, at full strength of interaction (i.e. t=4), we can get all exhaustive combination. In this case, the exhaustive combinations would be $2^4 = 16$.

As highlighted earlier, considering all exhaustive interaction is infeasible for large number of parameters and values. The next sub-sections demonstrate the fact that by adopting the t-way strategies (i.e. relaxing the interaction strength), the test data for testing can be systematically reduced. In this case, a step-by-step example will be demonstrated to illustrate the possible use of t-way strategies including uniform interaction, cumulative interaction, variable strength interaction, and input output relation based interaction.

Table 1. Base Data Values

Base Values	Input Variables			
	A	B	C	D
	a1	b1	c1	d1
a2	b2	c2	d2	

Table 2. Exhaustive Combination

Base Values	Input Variables			
	A	B	C	D
	a1	b1	c1	d1
a2	b2	c2	d2	
All Combinatorial Values	a1	b1	c1	d1
	a1	b1	c1	d2
	a1	b1	c2	d1
	a1	b1	c2	d2
	a1	b2	c1	d1
	a1	b2	c1	d2
	a1	b2	c2	d1
	a1	b2	c2	d2
	a2	b1	c1	d1
	a2	b1	c1	d2
	a2	b1	c2	d1
	a2	b1	c2	d2
	a2	b2	c1	d1
	a2	b2	c1	d2
	a2	b2	c2	d1
	a2	b2	c2	d2

3.1 Uniform Strength T-Way Interaction

Here, it is assumed that the interaction of variable is uniform throughout. Revisiting Table 1, and considering t=3, Fig. 2 highlights how the reduction is achieved. Firstly, the interaction is broken down between parameters ABC, ABD, ACD, and BCD.

Here, when parameters ABC are considered, the values for parameter D are don't cares (i.e. any random valid values for parameter D suffice). Similarly, when parameters ABD are considered, values for parameter C are don't cares. When parameters ACD are considered, values for parameter B are don't care. Finally, when parameters BCD are considered, values for parameter A are don't cares. Combining these results, we note that there are some repetitions of values between some entries for ABC, ABD, ACD and BCD. If these repetition is removed, we can get all the combinations at t=3.

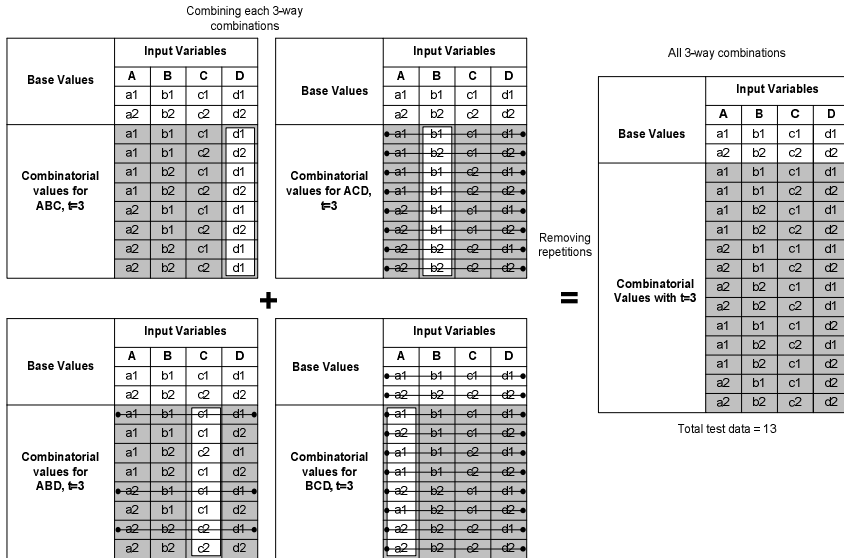


Fig. 2. Uniform t-way Interaction Results (t=3), CA (13,3,2⁴)

Here, we note that the test suite has been reduced from 16 (for exhaustive combination) to 13 (for t=3), a saving of 18.75 percent. Using the notation discussed earlier, we can write this test suite as $T_{suite} = CA (13,3,2^4)$.

3.2 Variable Strength T-Way Interaction

In many real applications, interaction may not be uniform for all parameters. Here, a particular subset of variables can have a higher interaction dependency than other variables (indicating failures due to the interaction of that subset may have more significant impact to the overall system). For example, consider a subset of components that control a safety-critical hardware interface. We want to use stronger coverage in that area (i.e. t=3). However, the rest of our components may be sufficiently tested with t=2. In this case, we can assign variable coverage strength to each subset of components as well as to the whole system.

To illustrate variable strength t-way interaction, we adopt the same example as Table 1. Now, we assume that all interaction is uniform at t=2 for all parameters (i.e. based on our result in Fig. 3). Then, we consider t=3, only for parameters B,C,D. Combining both interactions yield result shown in Fig. 3. Here, the test suite has been

reduced from 16 (for exhaustive case) to 13, a saving of 18.75 percent. Using the notation describe earlier, we can write this reduction as $T_{suite} = VCA(13,2,2^4, \{CA(3,2^3)\})$.

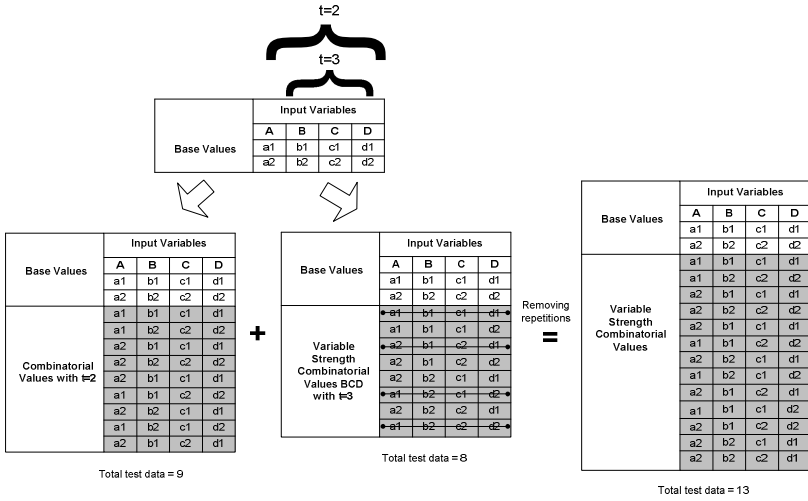


Fig. 3. Variable Strength Interaction, VCA (13,2,2⁴, {CA(3,2³)})

As a special case for VCA, we can also consider cumulative strength, t=3 and t=2. Revisiting Table 1, we can derive the test suite for t=2 using the same technique as t=3 (see Fig. 4).

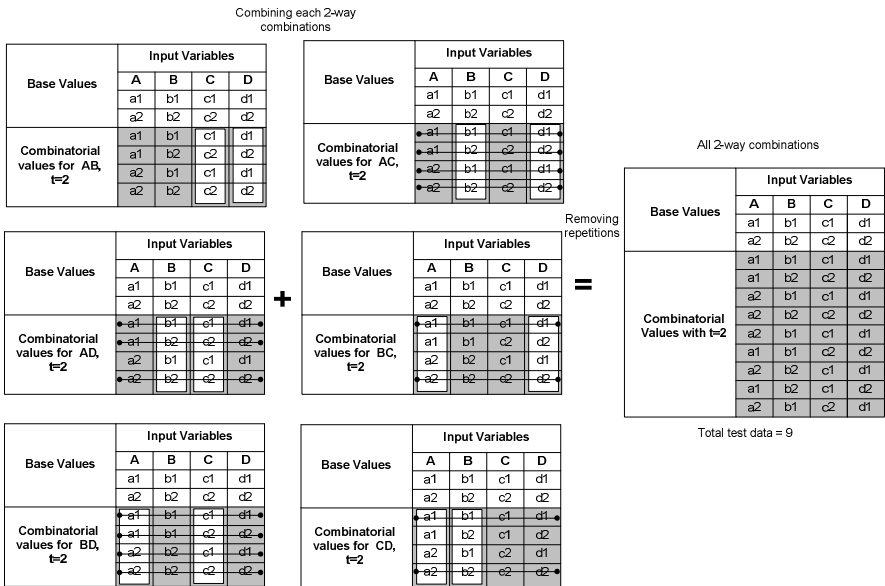


Fig. 4. Uniform t-way Interaction Results (t=2), CA (9,2,2⁴)

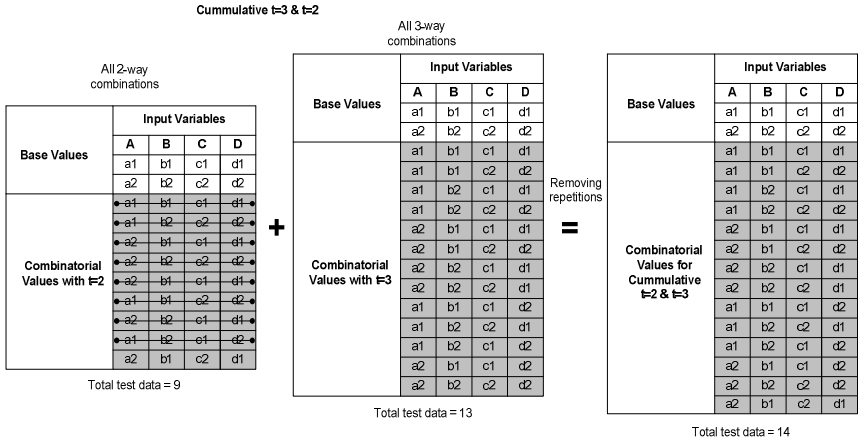


Fig. 5. Cumulative t=2 & t=3 Results, CA (14, {CA (9,2, 2⁴), {CA (13, 3, 2⁴)})
 Combining the test suite with t=3, yields the following result (see Fig. 5). Here, we note that T_{suite} for t=2 is not necessarily a subset of T_{suite} for t=3. In this case, the test suite has been reduced from 16 (for exhaustive case) to 14, a saving of 12.5 percent. Using the notation described earlier, we can write this reduction as T_{suite} = CA(9,2,2⁴) + CA (13,3,2⁴) or simply T_{suite} = CA (14, {CA (9,2, 2⁴), {CA (13, 3, 2⁴)}).

3.3 Input Output Relation Based Interaction

Similar to variable strength t-way interaction, input output relation based interaction does not deal with uniform interaction. Also, unlike other interaction possibilities discussed earlier, the reduction is performed by considering the knowledge on the input and output relationship amongst the parameter values involved. Normally, this relationship can be derived based on some statistical analysis such as Design of Experiments (DOE).

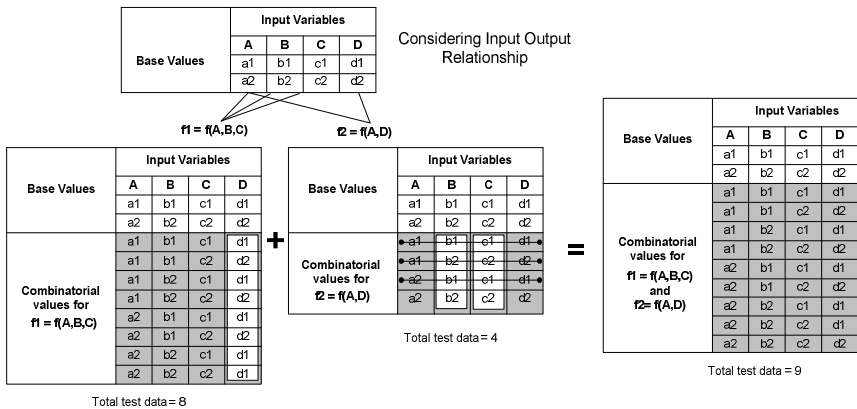


Fig. 6. Input Output Based Interaction, T_{suite} = IOR (9, 2⁴, {{0,1,2},{0,3}})

To illustrate the input output based interaction, we revisit Table 1 with the following input output relationship.

- i. Only two outputs are considered, f_1 and f_2 .
- ii. f_1 is a function of A,B,C, that is, $f_1=f(A,B,C)$.
- iii. f_2 is a function of A,D, that is, $f_2=f(A,D)$.

Ideally, these input output relationship are not to be assumed as they come from experimental results. Upon establishing these assumptions, we can derive the test suite accordingly. Fig. 6 illustrates the complete results. Here, the test suite has been reduced from 16 (for exhaustive case) to 9, a saving of 43.75 percent. Using the notation described earlier, we can write $R = \{\{0,1,2\}, \{0,3\}\}$ or the overall reduction as $T_{\text{suite}} = \text{IOR}(9, 2^4, \{\{0,1,2\}, \{0,3\}\})$.

4 Observation and Issues

The main observation here is the fact that by relaxing the interaction, we can systematically reduce the test data for consideration significantly. Other subtle observations can also be discussed further here.

- The final results for all cases discussed earlier (i.e. for uniform interaction, cumulative interaction, variable strength interaction, and input output relation based interaction) are not the most optimum, that is, all the interaction elements appear more than once. In order to be optimum, more efficient algorithms are required (see section 5 on related work).
- Uniform strength strategies are useful when there is little knowledge on the system under test. Thus, the interaction is assumed to be uniform throughout through judicious (e.g. based on experience on similar system) selection of interaction strength (t).
- Variable strength interaction strategies are applicable when a particular subset of input parameters has a higher interaction dependency than other parameters (indicating failures due to the interaction of that subset may have more significant impact to the overall system). As such, at least two interaction strengths can be assigned accordingly with one being stronger than the other.
- Input output relation based strategies are useful when the relationship amongst inputs and outputs are known (e.g. often through statistical method such as Design of Experiments) in advanced. Thus, the interaction can be properly established.

5 Related Works

The main aim of any t-way strategies is to ensure that the generated test suite covers the interaction tuples of interest for a certain type of interaction at least one whilst reducing the test data into manageable ones. However, there is no unique solution to this problem (i.e. NP-hard problem [5, 6]). In fact, it is unlikely that a unique strategy exists that can always generate the most optimum number of test case in every configuration.

A number of useful strategies have been developed from the last decade. A significant number of work have focused on pairwise ($t=2$) strategies (e.g. OATS (Orthogonal Array Test System) [7], IRPS [8], AllPairs [9], IPO [10], TCG (Test Case Generator) [11], OATSGen [12], ReduceArray2 [13], DDA (Deterministic Density Algorithm) [14], CTE-XL [15], rdExpert [16], and SmartTest [17]). As interaction is limited to $t=2$, pairwise strategies often yield the most minimum test set as compared other interaction. Although useful in some class of system, pairwise testing is known be ineffective for system with highly interacting variables [18-20]. For this reason, rather than dwelling on pairwise strategies, we are more interested on a general strategy for t -way test generation including that of variable strength, and input output based relations. The survey of each of these strategies is discussed next.

Klaib and Zamli developed a deterministic t -way strategy called GTWay [21, 22]. The main feature of GTWay is that it supports both test data generation and automated execution. This strategy heavily depends on its pair generation and backtracking algorithm. Once all pairs are generated, the backtracking algorithm will iteratively traverses all pairs in order to combine pairs with common parameter values in order to complete a test suite. To ensure the optimality of test data generated, combination of pairs can only be accepted if its cover the most uncovered pairs. In case of pairs that cannot be combined, the algorithm falls back to the first defined value.

Hartman et al. developed a t -way strategy, called IBM's Intelligent Test Case Handler (WHITCH), as Eclipse Java plug-in tool [23]. WHITCH uses the sophisticated combinatorial algorithms based on exhaustive search to construct test suites for t -way testing. Although useful as part of IBM's automated test plan generation, WHITCH results appear to be not optimized as far as the number of generated test cases is concerned. Furthermore, due to its exhaustive search algorithm, WHITCH execution times typically take a long time.

Jenkins developed a deterministic t -way generation strategy, called Jenny [24]. Jenny adopts a greedy algorithm to produce a test suite in one-test-at-a-time fashion. In Jenny, each feature has its own list of t -way interaction. It starts out with 1-way interaction (just the feature itself). When there are no further 1-way interaction left to cover, Jenny goes to 2-way interactions (this feature with one other feature) and so on. Hence, during generation instance, there could have one feature still covering 2-way interaction while another feature is already working on 3-way interactions. This process goes on until all interactions are covered.

Cohen et al developed the first commercialized t -way strategy, called AETG [25]. AETG starts with the generation of all possible parameter interactions. Based all the possible parameter interactions, AETG then decides the combination of values to maximize the interaction coverage so that it can build an efficient test set. This selection process is performed "one-test-at-a-time" until all the parameter interactions are covered. To enhance its capability (e.g. for better test size), a number of variant AETG implementations have been implemented such as that of mAETG [3], TCG [11] and mTCG [3].

Lei et al developed IPOG [26] based a novel "one-test-at-a-time" approach. In IPOG, the interaction parameters will be generated first as the partial test suite based on the number of parameters and interaction value. The test suite is then extended with the values of the next parameters by using horizontal and vertical extension mechanism. Here, horizontal extension extends the partial test suite with values of the next parameter to cover the most interaction. Upon completion of horizontal extension, vertical extension may be summoned to generate additional test cases that cover all

uncovered interactions. More recently, a number of variants have been developed to improve the IPOG's performance (i.e. IPOG-D [27], IPOF and IPOF2 [28]).

Younis and Zamli proposed another variant for IPOG named MIPOG [29, 30]. Addressing the dependency issue arising in IPOG strategy (i.e. generation of a test data can be unstable in IPOG due to the possibility of changing values during the vertical extension especially for test cases that include "don't care" value), MIPOG introduces two new algorithms for both horizontal and vertical extension. Here, the both algorithms remove the inherent dependency between subsequently generated test data (as occurred in IPOG family). Furthermore, MIPOG strategy also further optimizes the don't care value during vertical extension making this strategy in most cases outperform IPOG in term of test size. As the data dependency issue is removed, Younis and Zamli implemented the parallel MIPOG strategy in the Multi-core platform (called MC-MIPOG (MultiCore MIPOG)) [31] as well as in the Grid platform (called G-MIPOG (Grid MIPOG)) [32]. By implementing the strategy in multi-core and grid environment, the time taken to produce the final test suite for MIPOG strategy is reduced.

Arshem developed a freeware Java based t-way testing tool called Test Vector Generator (TVG) [33] based on extension of AETG strategy to support t-way testing. Similar efforts are also undertaken by Bryce and Colbourn [34, 35] to enhance AETG for t-way testing. Nie et al. [36] proposed a generalization for IPO with genetic algorithm (GA), called IPO_N, and GA_N respectively for $t=3$. Here, IPO_N performed better than GA_N in terms of test size as well as execution time [36].

As far as variable strength t-way strategies are concerned, Cohen et al. implemented the first model t-way strategy with variable strength based capability based on simulated annealing (SA) [37]. Although generating optimal test suites, this approach is very time consuming because all interaction elements needs to be analyzed exhaustively using binary search strategy.

Wang et al. extended the model proposed by Cohen et al [37] and proposed a more general strategy relying on two greedy algorithms. The first algorithm is based on one-test-at-a-time strategy while the other algorithm is based on in-parameter-order strategy [38]. Although useful as far as addressing the limitation of the Cohen's model in terms of the need for the interaction strength (t) involved to be disjoint, Wang et al. approach appears to produce non-optimized set for mixed parameter values.

Chen et al. proposed a variant algorithm based on ant colony approach (named Ant Colony Strategy (ACS)) in order to support variable strength capability [39]. Similar to Cohen et al [37], this approach is also time consuming and supports low interaction strength $2 \leq t \leq 3$. Apart from these approaches, new version of TVG [33] also addresses the variable strength capabilities but with non-optimized set.

Concerning the strategies that address the support for input output based relations much work has started to appear. The input output based strategies can be considered as the general case for t-way strategies as they can be customized to behave as such to support all interaction possibilities (i.e. uniform strength and variable strength interactions). However, if the parameters are large, setting up for uniform and variable strength interaction can be cumbersome as there is a need to define all the relations for each interaction.

Schroeder and Korel developed an input output relations based strategy called Union [40, 41]. In the case of Union, the strategy generates the test suite for each output variable that cover all associated input interaction and then assign random

value for all the ‘don’t care’. Then, the strategy finds the union of all test suites in order to reduce the number of generated test data.

Building from the Union Strategy, Schroeder et al developed the Greedy strategy [41, 42]. Similar to Union, the Greedy strategy also generates the initial test suite that covered all associated input interaction by randomly selecting values for all don’t care parameters. Nonetheless, unlike the Union strategy, the Greedy strategy picks only the unselected test case from the initial test suite which covers the most uncovered interactions as the final test suite. In this manner, the Greedy strategy often generates a more optimal test size than that of the Union Strategy.

Wang et al developed two strategies to support input based relations called ParaOrder and ReqOrder [43]. ParaOrder strategy implements horizontal and vertical extension for generating the final test cases, much like the uniform strength IPOG implementation [26]. The main difference between ParaOrder with IPOG is the fact that the initial test case for the former is generated based on the first defined input output relationships while the initial test case for the latter is generated in-defined-order-of-parameter found. In the case of ReqOrder, the selection of initial test case does not necessarily follow the first defined input output relationships rather the selection is done based on the highest input output relationship coverage.

6 Analysis of Related Works

In order to help test engineers make inform decision on the adoption of a particular t-way strategy implementation, Table 3 provides the digest information regarding the supported interactions by all strategies discussed earlier. It should be noted that an “√*” indicates that the strategy implementation of interest partially supports the said interaction (i.e. only supports pairwise interaction) whereas an “√” indicates that the strategy implementation of interest provides full support. An “X” indicates the missing support.

Table 3. Analysis of Related Works

Strategy	Uniform Strength	Variable Strength	I/O Relations	Strategy	Uniform Strength	Variable Strength	I/O Relations
OATS	√*	X	X	IPOG	√	X	X
IRPS	√*	X	X	IPOG-D	√	X	X
AllPairs	√*	X	X	IPOF	√	X	X
IPO	√*	X	X	IPOF2	√	X	X
TCG	√*	X	X	MIPOG	√	X	X
OATSGen	√*	X	X	MC-MIPOG	√	X	X
ReduceArray2	√*	X	X	G-MIPOG	√	X	X
DDA	√*	X	X	TVG	√	√	√
CTE-XL	√*	X	X	IPO_N	√	X	X
rdExpert	√*	X	X	GA_N	√	X	X
SmartTest	√*	X	X	SA	√	√	X
GTWay	√	X	X	ACS	√	√	X
WHITCH	√	X	X	Union	√	√	√
Jenny	√	X	X	Greedy	√	√	√
AETG	√	X	X	ParaOrder	√	√	√
mAETG	√	X	X	ReqOrder	√	√	√
mTCG	√	X	X				

Referring to Table 3, most strategy implementations merely support uniform strength interactions. Here, only Union, TVG, ParaOrder, and ReqOrder support all possible types of interactions. With such knowledge, test engineers can adopt the tool implementation accordingly (i.e. based on the interaction requirements).

7 Conclusion

Summing up, this paper has presented the different possible uses of t-way strategies including uniform interaction, variable strength interaction, and input-output based relations. Additionally, this paper has also analyzed the related works by highlighting the capabilities of some of the existing t-way strategy implementations. It is hoped that such an analysis can help test engineers choose the interaction type of interest as well as the tool support required.

Finally, while much useful research work has been done in the last decade (i.e. as evident by the large number of developed strategy implementations), the adoption of interaction testing for studying and testing real life systems has not been widespread [44]. In order to address this issue, more research into the algorithms and techniques are required to facilitate its adoption in the main stream of software engineering.

Acknowledgement

This research is partially funded by the generous MOHE fundamental grants – “Investigating T-Way Test Data Reduction Strategy Using Particle Swarm Optimization Technique” and USM RU grants – “Development of Variable-Strength Interaction Testing Strategy for T-Way Test Data Generation”.

References

- [1] Zamli, K.Z., Younis, M.I., Abdullah, S.A.C., Soh, Z.H.C.: Software Testing, 1st edn. Open University, Malaysia KL (2008)
- [2] Kuhn, D.R., Lei, Y., Kacker, R.: Practical Combinatorial Testing: Beyond Pairwise. *IEEE IT Professional* 10(3), 19–23 (2008)
- [3] Cohen, M. B.: Designing Test Suites For Software Interaction Testing. PhD Thesis, School of Computer Science, University of Auckland (2004)
- [4] Zekaoui, L.: Mixed Covering Arrays On Graphs And Tabu Search Algorithms. Msc Thesis, Ottawa-Carleton Institute for Computer Science, University of Ottawa, Ottawa, Canada (2006)
- [5] Shiba, T., Tsuchiya, T., Kikuno, T.: Using Artificial Life Techniques To Generate Test Cases For Combinatorial Testing. In: Proceedings of the 28th Annual Intl. Computer Software and Applications Conf. (COMPSAC 2004), Hong Kong, pp. 72-77 (2004)
- [6] Younis, M.I., Zamli, K.Z., Klaib, M.F.J., Soh, Z.H.C., Abdullah, S.A.C., Isa, N.A.M.: Assessing IRPS As An Efficient Pairwise Test Data Generation Strategy. *International Journal of Advanced Intelligence Paradigms* 2(3), 90–104 (2010)
- [7] Krishnan, R., Krishna, S.M., Nandhan, P.S.: Combinatorial Testing: Learnings From Our Experience. *ACM SIGSOFT Software Engineering Notes* 32(3), 1–8 (2007)

- [8] Younis, M.I., Zamli, K.Z., Isa, N.A.M.: IRPS: An Efficient Test Data Generation Strategy For Pairwise Testing. In: Proceedings of the 12th International Conference on Knowledge-Based Intelligent Information and Engineering Systems, Part I, pp. 493–500. Springer, Zagreb, Croatia (2008)
- [9] Allpairs Test Case Generation Tool, <http://www.satisfice.com/tools.shtml>
- [10] Lei, Y., Tai, K.C.: In-Parameter-Order: A Test Generation Strategy For Pairwise Testing. In: Proceedings of 3rd IEEE International Conference on High Assurance Systems Engineering Symposium, Washington DC, USA, pp.254–261 (1998)
- [11] Tung, Y.W., Aldiwan, W.S.: Automatic Test Case Generation For The New Generation Mission Software System. In: Proceedings of IEEE Aerospace Conference, pp. 431–437. Big Sky, MT, USA (March 2000)
- [12] Harrell, J.M.: Orthogonal Array Testing Strategy (OATS) Technique: Seilevel, Inc. (2001)
- [13] Daich, G.T.: Testing Combinations Of Parameters Made Easy (Software Testing). In: Proceedings of IEEE Systems Readiness Technology Conference (AUTOTESTCON 2003), pp. 379–384 (2003)
- [14] Colbourn, C.J., Cohen, M.B., Turban, R.C.: A Deterministic Density Algorithm For Pairwise Interaction Coverage. In: Proceedings. of the Intl. Conference on Software Engineering (IASTED 2004), pp. 345–352 (2004)
- [15] Lehmann, E., Wegener, J.: Test Case Design By Means Of The CTE-XL. In: Proceedings of the 8th European International Conference on Software Testing, Analysis & Review (EuroSTAR 2000), Copenhagen, Denmark (2000)
- [16] Copeland, L.: A Practitioner's Guide To Software Test Design. Massachusetts, STQE Publishing, USA (2004)
- [17] SmartTest - Pairwise Testing, <http://www.smartwaretechnologies.com/smartistestprod.htm>
- [18] Kuhn, D.R., Wallace, D.R., Gallo, A.M.: Software Fault Interaction And Implication For Software Testing. IEEE Transaction on Software Engineering. 30(6), 418–421 (2004)
- [19] Younis, M.I., Zamli, K.Z.: Assessing Combinatorial Interaction Strategy For Reverse Engineering Of Combinational Circuits. In: Proceedings of the IEEE Symposium on Industrial Electronics and Applications (ISIEA 2009), Kuala Lumpur, Malaysia (2009)
- [20] Younis, M.I., Zamli, K.Z.: A Strategy for Automatic Quality Signing And Verification Processes For Hardware And Software Testing. Advances in Software Engineering 1–7 (2010)
- [21] Zamli, K.Z., Klaib, M.F.J., Younis, M.I., Isa, N.A.M., Abdullah, R.: Design And Implementation Of A T-Way Test Data Generation Strategy With Automated Execution Tool Support. Information Sciences 181(9), 1741–1758 (2011)
- [22] Klaib, M. F. J.: Development Of An Automated Test Data Generation And Execution Strategy Using Combinatorial Approach. PhD. Thesis, School of Electrical And Electronics, Universiti Sains Malaysia (2009)
- [23] IBM Intelligent Test Case Handler, <http://www.alphaworks.ibm.com/tech/whitch>
- [24] Jenny Test Tool, <http://www.burtleburtle.net/bob/math/jenny.html>
- [25] Cohen, D.M., Dalal, S.R., Fredman, M.L., Patton, G.C.: The AETG System: An Approach To Testing Based On Combinatorial Design. IEEE Transactions on Software Engineering 23(7), 437–444 (1997)
- [26] Lei, Y., Kacker, R., Kuhn, D.R., Okun, V., Lawrence, J.: IPOG: A General Strategy For T-Way Software Testing. In: Proceedings of the 14th Annual IEEE International Conference and Workshops on The Engineering of Computer-Based Systems, Tucson, AZ, pp. 549–556 (2007)
- [27] Lei, Y., Kacker, R., Kuhn, R., Okun, V., Lawrence, J.: IPOG/IPOG-D: Efficient Test Generation For Multi-way Combinatorial Testing. Journal of Software Testing, Verification and Reliability 18(3), 125–148 (2008)

- [28] Forbes, M., Lawrence, J., Lei, Y., Kacker, R., Kuhn, D.R.: Refining The In-Parameter-Order Strategy For Constructing Covering Arrays. *Journal of Research of the National Institute of Standards and Technology*. 113(5), 287–297 (2008)
- [29] Younis, M.I., Zamli, K.Z., Isa, N.A.M.: MIPOG - Modification Of The IPOG Strategy For T-Way Software Testing. In: *Proceeding of The Distributed Frameworks and Applications (DFmA)*, Penang, Malaysia (2008)
- [30] Younis, M. I.: MIPOG: A Parallel T-Way Minimization Strategy For Combinatorial Testing. PhD. Thesis, School of Electrical And Electronics, Universiti Sains Malaysia (2010)
- [31] Younis, M.I., Zamli, K.Z.: MC-MIPOG: A Parallel T-Way Test Generation Strategy For Multicore Systems. *ETRI Journal* 32(1), 73–83 (2010)
- [32] Younis, M.I., Zamli, K.Z., Isa, N.A.M.: A Strategy For Grid Based T-Way Test Data Generation. In: *Proceedings the 1st IEEE International Conference on Distributed Frameworks and Application (DFmA 2008)*, Penang, Malaysia, pp. 73–78 (2008)
- [33] TVG, <http://sourceforge.net/projects/tvg>
- [34] Bryce, R.C., Colbourn, C.J.: A Density-Based Greedy Algorithm For Higher Strength Covering Arrays. *Software Testing, Verification and Reliability* 19(1), 37–53 (2009)
- [35] Bryce, R.C., Colbourn, C.J.: The Density Algorithm For Pairwise Interaction Testing. *Software Testing, Verification and Reliability*. 17(3), 159–182 (2007)
- [36] Nie, C., Xu, B., Shi, L., Dong, G.: Automatic Test Generation For N-Way Combinatorial Testing. In: Reussner, R., Mayer, J., Stafford, J.A., Overhage, S., Becker, S., Schroeder, P.J. (eds.) *QoSA 2005 and SOQUA 2005*. LNCS, vol. 3712, pp. 203–211. Springer, Heidelberg (2005)
- [37] Cohen, M.B., Gibbons, P.B., Mugridge, W.B., Colbourn, C.J., Collofello, J.S.: Variable Strength Interaction Testing Of Components. In: *Proceedings of 27th Annual International Computer Software and Applications Conference*, Dallas, USA pp. 413–418 (2003)
- [38] Wang, Z., Xu, B., Nie, C.: Greedy Heuristic Algorithms To Generate Variable Strength Combinatorial Test Suite. In: *Proceedings of the 8th International Conference on Quality Software*, Oxford, UK, pp. 155–160 (2008)
- [39] Chen, X., Gu, Q., Li, A., Chen, D.: Variable Strength Interaction Testing With An Ant Colony System Approach. In: *Proceedings of 16th Asia-Pacific Software Engineering Conference*, Penang, Malaysia, pp. 160–167 (2009)
- [40] Schroeder, P.J., Korel, B.: Black-Box Test Reduction Using Input-Output Analysis. *SIGSOFT Software Engineering Notes* 25(5), 173–177 (2000)
- [41] Schroeder, P. J.: Black-Box Test Reduction Using Input-Output Analysis. PhD Thesis, Department of Computer Science, Illinois Institute of Technology, Chicago, IL, USA (2001)
- [42] Schroeder, P.J., Faherty, P., Korel, B.: Generating Expected Results For Automated Black-Box Testing. In: *Proceedings of 17th IEEE International Conference on Automated Software Engineering (ASE 2002)*, Edinburgh, Scotland, UK, pp. 139–148 (2002)
- [43] Wang, Z., Nie, C., Xu, B.: Generating Combinatorial Test Suite For Interaction Relationship. In: *Proceedings of 4th International Workshop on Software Quality Assurance (SOQUA 2007)*, Dubrovnik, Croatia, pp. 55–61 (2007)
- [44] Czerwonka, J.: Pairwise Testing In Real World. In: *Proceedings of 24th Pacific Northwest Software Quality Conference*, Portland, Oregon, USA, pp. 419–430 (2006)

On the Relationship between Proof Writing and Programming: Some Conclusions for Teaching Future Software Developers

Michael Hartwig

Multimedia University, Malaysia
Michael.Jua.Hartwig@gmail.com

Abstract. The analogy between proving theorems and writing computer programs has been debated for a long time. In a recent paper, Calude and others [5] argue that - albeit mentioned analogy seems to exist - the role of proof in mathematical modeling (and hence programming) is very small. Describing the act of proving and the act of writing a computational program with the help of the SECI model, a model used widely to describe knowledge creation processes, it can be argued that the thought processes needed for both activities complement each other. This fact can then be used to justify a sound and rigorous training in proof writing for the programmer and future software developer.

Keywords: proof writing, programming, software development, computer science education.

1 Motivation

The similarity (or difference) between the act of proving a theorem and the act of writing a computer program has caused many discussions in the computer science community [1,2]. Some authors like Friske [3] focussing on the immanent use of abstract objects in both activities find a strong similarity: “There is a close analogy between the thought processes used in computer programming and the thought processes for proof writing.” Daly [4] described similar experiences in teaching concluding: “Constructing a mathematical proof is isomorphic with writing a computer program. Validating a proof is as difficult as validating a computer program.”

Others like Calude [5] stressed the importance of the outcome of the thought processes involved. In their view, computer programs correspond to models and proofs correspond to algorithms. Henceforward, computer programs are subject to adequacy tests, while theorems and proofs are subject to a correctness test. This difference is often justified by the (de facto) non existence of correctness proofs in the programming world. But it also allows for a different view of programming where programming is seen as an art that is centred around human machine interaction. It could then be easily interpreted to minimize or relegate the role of mathematics and abstract thinking taught in computer science classes

even further. It is obvious that this debate, now going on for more than a decade, has serious consequences for the way we see our science and the development of appropriate curricula. Duncan [6] concludes in a response to Crowcroft [1]: "The age-old art versus science versus engineering debate is still not settled." (See also Dennings interesting reaction in [7].)

There is a little truth in everything. A program (the outcome of the programmers activity) has to be looked at from different viewpoints to address the different needs of its users. A program should do what the user expects it to do (the problem solving or engineering view), it should contain as few as possible errors (the correctness or scientific view), it should be user friendly (the human-machine interaction view), a joy to work with (the art view), and probably adhere to many more standards not discussed here. It is therefore to be expected that the act of programming comprises a broad range of different and sometimes interfering or contrary activities giving reasons for above mentioned discussion.

The paper tries to look at the problem solving or engineering view of programming and contribute to an improvement in its teaching. As already mentioned, there has been presented some evidence suggesting that an improvement in the skills needed to write a proof may improve also the proof writers programming skills. See again [3,4] for some studies. However, such studies will not completely reveal the causes why a certain programmer or group of programmers was able to solve a given programming task because of the complex nature of the act of software development itself that depends on analytical, creative, linguistic, and other skills.¹ A second difficulty can be seen in the fact that changes in teaching aimed at improving the programming skills of the students usually consist of more than only an intensified use of proofs.

Nevertheless, above studies coincide with our experiences made at Multimedia University (MMU). In 2002 students were encouraged for the first time to participate in the regional "eGenting Programming Competition"². Unfortunately, the achievements of our students were disappointing. MMU had only one graduate students winning a third place and one graduate winning a merit prize. Based on the results the Faculty of Information Technology implemented some changes in the introductory programming courses CP1, CP2 and the Discrete Structure course DS. While no changes were done in terms of the syllabus, CP1 and CP2 started to stress the importance of structured programming and the writing of readable and neat code while at the same time adding more intermediate tests.

¹ During the 5th Conference of the Association of Commonwealth Examinations and Accreditation Bodies an author presented an interesting example demonstrating that in some provinces of South Africa students were under-performing in a mathematical exercise not because of its difficulty but because of the fact that the task was centred around calculating the number of slices of a pizza and the majority of mentioned students did not have seen a pizza before!

² Information about the competition (Exercises, solutions, winners, photos) can be found under <http://www.genting.com.my>. The competition can be considered the most difficult programming competition in the region. In its first years winners were always be given a job guarantee.

The DS course changed its methodology from a "teaching by example" style to an approach using proofs and abstract, formal definitions as often as possible. Such changes were not welcomed by all members of staff wholeheartedly. The CP team had to struggle with complains about the higher workload while the DS team had to convince colleagues that teaching in such a way might be challenging but may indeed support programming skills.

In the following year MMU had again no student winning a top prize at eGenting; one graduate student were mentioned as a winner of a Distinct Prize. However, 4 out of 6 merit prizes were won by our undergraduate students who were, in general, also outperforming our senior students. It was a logical consequence that in the following year MMU claimed its first victory, won the First Runner Up Prize and Distinct and Merit prizes. Using above mentioned experiences the paper is then aimed at contributing to the study of the relationship between proof writing and programming trying to give first answers to the following questions.

1. What could be the basis of a study of the relationship between both activities?
2. Why could the skills trained in one area support the development of skills in the other?
3. How can this relationship be investigated further?
4. How can the results be interpreted and implemented in our teaching?

As argued in the following: Programming (seen from an engineering or problem solving perspective) and mathematical proof writing can be understood as complementary activities in a knowledge creation process. This demonstrates that both activities indeed share some similarities and that teaching one of them increases skills in the other. It should also provide arguments and directions for further studies of the relationship between proof writing on one side and programming on the other. Finally, some suggestions for using this new understanding in the teaching of young programmers are given.

2 The SECI Model

Armour [8,9,10,11] consistently argues that programming and software development should be understood as a knowledge development process. It might therefore be interesting to consider proof writing in the same way – as an activity in which we learn and create knowledge. One of the common models used to understand the activities related to knowledge creation and acquisition has been developed by Nonaka and Takeuchi [12]. Called SECI (after its Socialization, Externalization, Combination, and Internalization steps), the model describes the processes involved in form of a spiral of interleaving steps of activities related to externalization and internalization (see figure 1).

Externalization comprises hereby all activities transforming tacit into explicit knowledge. An example might be the translation of quantitative research

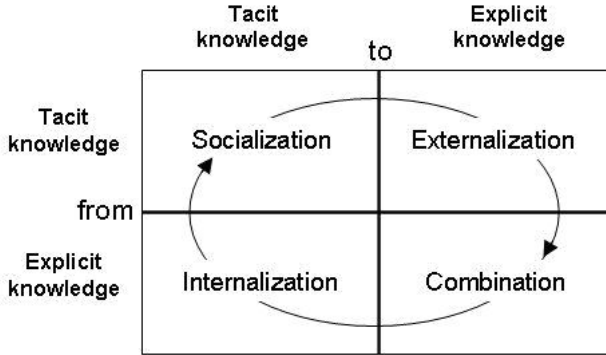


Fig. 1. The SECI model (Source: <http://www.hcklab.org/research/knowledgemanagement/tacit-explicit-knowledge.htm>)

data into a recommendation for practice. *Internalization* transforms explicit into tacit knowledge, usually a practical activity in which we follow rules or procedures observing the effects to develop personal experience. *Combination* and *Socialization* processes do not alter the status of the knowledge (whether it is tacit or explicit) but comprise activities where the knowledge is shared, applied, discussed, communicated, and published. While *Combination* and *Socialization* are also processes in which we learn, they will be considered as intermediate steps in the following. The SECI model has been applied to a range of different areas [13,14,15].

3 Programming and Proving

Programming can be seen simply as a way to solve problems with computers. It can involve a very broad range of activities including algorithm development, translation actions (such as "coding"), implementation, and testing. But it can also include the design of user interfaces, system-system interfaces and the creation of database systems, to name just a few abstract design activities. As mentioned already, Armour [8,9,10,11] sees all those actions as more or less concrete knowledge development activities. That is, activities that lead to the development of a more-or-less adequate (or correct) model – a model that simulates a virtual or real-world analogous environment and solves problems within that environment. Calude states that algorithms implemented in these models are rarely accompanied by a proof of correctness. Whether this could be seen as a sign for the declined role of proofing in programming or not, has yet to be decided.

Nevertheless, it could be asserted that programming is the art of developing "adequate" models and simulating or instantiating them with a computer program (figure 2). In other words, a programmer has to perform tasks that could be defined as

- activities that try to build adequate models for a given problem (situation) or set of data inputs and outputs, and
- activities that translate (or: communicate) those models in an appropriate manner to a computer in a way that "assures" correct simulation.

Fig. 2. Mental skills relevant to programming

On the other hand, *proofs* may be seen as a tool to demonstrate or establish knowledge with the objective of convincing somebody. Given the strictness of proof approaches, a proof writer can be considered as a person assuming unproven principles (axioms) and applying deduction or inference rules in an intelligent and creative way that will allow him to reach a desired conclusion. While such strict proofs are named Hilbertian or monolithic and, as Calude points out, ideally sound, it is hard to find such proofs in mathematical books and articles. Still, the proofs there could be turned in to such a form as every newly gained conclusion follows in a "convincing" [16] manner from the set of conclusions obtained so far.

Without doubt, programs and also proofs will undergo major changes in future. However, and following above thoughts activities related to proof writing (figure 3) can be clearly characterised as

- activities that try to deduct properties from a given model using established or accepted rules, and
- activities that communicate those deductions in a meaningful and convincing manner to others.

Fig. 3. Mental skills relevant to proof writing

Having characterised the thought processes required for writing programs and proofs in above manner it is evident that activities related to *programming and model building* in general with an emphasis on adequacy would correspond to the SECI models *internalization step*, defined as a practical, tacit knowledge gaining process. A model is developed within the programmer's mind and, after several steps of trial and error, he or she may have a "feeling" of what ought to be the most correct (or adequate) description of the world to simulate – a description that may be very different from the model he or she had first in mind. This contrasts with *proof writing* activities which clearly correspond to Takeuchi and Nonaka's *externalization step*. Before starting to write a proof, the proof writer knows already the desired outcome (represented by the hypothesis) and is only concerned with an explicit, correct, and traceable description verifying the

Table 1. Relating mental skills relevant to proof writing and programming to steps in the SECI model

SECI Model	Mental skills relevant to programming	Mental skills relevant to proof writing
Internalization	build adequate models	
Socialization	translate (communicate) obtained models to a computer	
Externalization		deduct properties from models (or programs)
Combination		communicate those deductions to others or prepare for re-use in new models

correctness of the hypothesis. At its core, this is an ideal transformation of tacit (or believed-to-be-true) knowledge into explicit (ready-to-be-shared) knowledge.

So far above arguments are still in line with the thoughts mentioned already by Calude and his colleagues [5]. They might also be used to support a declining role of proof writing in the programming world. The thought processes between programming and proving seem to be different, if not contrary. However, and as argued in the following, this is only true if having a very narrow view of a programmers and proof writers daily activities. A view that does not correspond to recent developments.

Problems and tasks in today's computer science as well as state-of-the-art business programming projects require both, the computational theorist and the programmer, to equally master the skills described so far. As mentioned earlier, the aim of programming is to model adequately a representation of a scene of the world. During this knowledge acquisition and complex process usually sub-models have to be developed and separately tested to determine whether its properties and behaviour matches those needed and whether it can be integrated into the system. Whenever the writer of a program is testing or debugging such sub-models, he/she is reasoning about properties of it (and the program). The outcomes might not be formalised as a conventional proof, but the programmer must develop an internal conviction about certain properties of the model. For example, even the quick check whether the bounds of a programmed loop are set correctly could be considered as a kind of property deduction activity. More complex and important requirements such as the space and time used by parts of the program will require the programmer to reason deeply about the code. Although this might be done using test scenarios alone, the set up of these scenarios necessarily depends upon properties of the program.

During this process of improving (sub-)models the program writer might even deduce new properties that can be added to the project's objectives and could give the software development process completely new directions. It is the nature

of any knowledge acquisition process to include phases of (what Armour calls) "Second Order Ignorance" (2OI). Situations in which we do not know that there is something we do not know [10]. Resolving 2OI requires a different process that needs to be intersected with reflective (debugging, proving) steps able to provide guidance and new directions. Modern software development methodologies such as agile methods reflect this justifying Daly's conclusion:

"Mathematical problem solving is a significant part of the programming process." [4]

Similar thoughts apply to proof writing. Writing a proof is a knowledge acquisition process in which the writer learns how to deduce properties. Seldom he/she is aware of the correct steps leading towards the hypothesis before writing the proof. Complex proofs then demand the scientist to constantly split the proof into sub-proofs and create models that could require demonstration of some of the required properties separately. Often the scientist is hereby forced to change the model and environment in which the problem was created in order to be able to understand and solve it. Taking it further: In many cases the most important things that are learned from proofs are not the deduced properties but rather the (sub-)models or techniques and their properties that had to be created during the proof writing process.

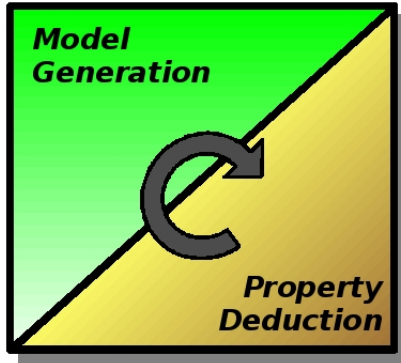


Fig. 4. Mental skills needed in the process of writing programs as well as proofs

Henceforward, programming and proof writing can be considered as complementary activities of interwoven traceable, convincing (or persuasive) steps of model building and property deduction (See figure 4). The similarity to the SECI model is appealing and justifies a rigorous training in both mental processes for the programmer and the proof writer as well. A further and thorough study of this relationship between both activities based on above findings is strongly recommended. Firstly, this research should study the linkage between proof writing and programming in selected programming areas (*Are we able to quantify*

or measure the amount of the programmers activities that we would classify as model building on one side and those classified as property deduction on the other? Is such a differentiation always possible?). Secondly, it should look at the direct impact of a stronger education in proof writing on the skills of young programmers.

4 Suggestions for Teaching

Above thoughts allow first conclusions. If programming and proof writing are activities producing outcomes that will be read by others and are made to convince others (either the computer or another scientist) there is one more reason to write modularized, readable and neat code. Students need to be reminded about it constantly.

There are also some activities that make a clear use of above mentioned relationship between proof writing and programming. If proof writing and programming consists of such interwoven steps then students should be trained in converting proofs and formal definitions into programs. Classes in discrete mathematics or set theory are ideal places to train such skills. Those courses provide the students with a variety of different lemmata, definitions and proofs that can be the basis for training a very close and direct translation of formal descriptions into programmed code. Students will quickly realize that a formal definition can indeed be converted into a (often recursive) procedure in a programming language quite straightforwardly. However, in those courses students are writing proofs only on paper or use complete mathematical packages for the display of new mathematical structures missing the great opportunity to demonstrate that all structures learned are of great value to the programmer as data structures on one hand and missing the chance to train the students programming skills on the other.

Knowing then that programming is in the same way an act always involving some steps that could be interpreted as proof writing students should also be trained in such activities more intensively and directly. In other words, students need to be asked to deduce properties of a given code. They need to learn to reason about code. Very typical question the student should then answer might include the following.

- What is the status of a selected variable at different run times?
- Can we find an instantiation of a specific variable that will cause the program to behave unintentionally?
- Are there any side effects?
- How much space, time will the program use?
- Could the problem at hand be solved with a program having less lines of code?
- Could the problem at hand be solved with a longer, but maybe more elegant solution?

While such questions might look as if they belong to every programming class, there are hard to be found. Calude and his colleagues are not only correct when referring to professional software development: Thorough and deep reasoning about written code is, unfortunately, also not found in most programming classes. Teaching time is spend on programming and not on reflection of the written code. Above characterisation of programming and proof writing seems to suggest that programming skills depend on this ability to reflect and deduce properties of written code, which is different from writing new code³. Training such skills might therefore have a higher impact on the programmers skills than considered now. Robertson's suggestions in *Unleash the inner computer scientist in everyone!*^[17] related to a creative and enjoyable learning environment were also not "rocket science". However, her mandate seems to go into the same direction and should be heard.

"What I'm suggesting is not new - there are pockets of excellence outreach work with kids in various parts of the world. I think it is time we tried more of it, even although it is time consuming." ^[17]

5 Conclusion

The thought processes in activities related to writing a mathematical proof and a computer program can equally be classified as knowledge acquisition activities. Seen from a historical perspective the mental activities that deduce properties from models can be classified as proving while the generation of adequate models can be classified as programming. The paper has provided some arguments to consider this a too restricted and also misleading view. Modern programs and complex proofs require the writer to be conversant in both skills, which in fact complement each other. Programming and proving can be defined as processes with intersecting model generation and property deduction steps. This clearly justifies a sound and rigorous training in mathematical sciences and proof writing and supports the education of good programmers and problem solvers. Further research based on those findings is recommended.

Acknowledgement

I would like to thank Philip G. Armour for commenting on a previous version of this article as well as all colleagues from the Faculty of Information Technology at the Multimedia University involved in the improvement of the programmes offered.

³ Every programmer will agree. It is often quite a challenge to understand code written by someone else which does not only seem to depend on the way the code is written alone. Understanding code written by someone else can be challenging and requires different intellectual abilities.

References

1. Crowcroft, J.: On the nature of computing. *Communications of the ACM* 48(2), 19–20 (2005)
2. Calude, C., Calude, E., Marcus, S.: Passages of Proof. *Bulletin of the EATCS* 84, 167–188 (2004)
3. Friske, M.: Teaching Proofs: A Lesson from Software Engineering. *The American Mathematical Monthly* 92, 142–144 (1985)
4. Daly, C., Waldron, J.: *Introductory Programming, Problem Solving and Computer Assisted Assessment* (2002)
5. Calude, C., Calude, E., Marcus, S.: *Proving and Programming*. CDMTCS Research Report Series 06 (2007)
6. Duncan, S.: Response to John Crowcroft. *Communications of the ACM* 48(4), 13 (2005)
7. Denning, P.J.: Is computer science science? *Communications of the ACM* 48(4), 27–31 (2005)
8. Armour, P.G.: Ten unmyths of project estimation. *Communications of the ACM* 45(11), 15–18 (2002)
9. Armour, P.G.: Beware of counting LOC. *Communications of the ACM* 47(3), 21–24 (2004)
10. Armour, P.G.: Not-defect: the mature discipline of testing. *Communications of the ACM* 47(10), 15–18 (2004)
11. Armour, P.G.: The unconscious art of software testing. *Communications of the ACM* 48(1), 15–18 (2005)
12. Nonaka, I., Takeuchi, H.: *The Knowledge Creating Company*. Oxford University Press, Oxford (1995)
13. Rice, J.L., Rice, B.S.: The applicability of the SECI model to multi organisational endeavours: an integrative review. *International Journal of Organisational Behaviour* 9(8), 671–682 (2005)
14. Kutay, C., Aurum, A.: *Validation of SECI Model in Education*. Technical Report 0524, The University of New South Wales, Sydney, Australia (2005)
15. Chatti, M.A., Klamma, R., Jarke, M., Naeve, A.: The Web 2.0 Driven SECI Model Based Learning Process. In: *ICALT*, pp. 780–782 (2007)
16. DeMillo, R.A., Lipton, R.J., Perlis, A.J.: Social Processes and Proofs of Theorems and Programs. *Communications of the ACM* 22(5), 271–280 (1979)
17. Robertson, J.: Computer science outreach: Meeting the kids halfway. *Commun. ACM* 10(10), 89 (2009)

Model-Based Web Components Testing: A Prioritization Approach

Ahmed Al-Herz* and Moataz Ahmed

Information and Computer Science Department,
King Fahd University of Petroleum and Minerals,
Dhaharan 31261, Saudi Arabia
{alherz, moataz}@kfupm.edu.sa

Abstract. Web applications testing and verification is becoming a highly challenging task. A number of model-based approaches has been proposed to deal with such a challenge. However, there is no criteria that could be used to aid practitioners in selecting appropriate approaches suitable for their particular effort. In this paper we present a set of attributes to serve as criteria for classifying and comparing these approaches and provide such aid to practitioners. The set of attributes is also meant to guide researchers interested in proposing new model-based Web application testing and verification approaches. The paper discusses a number of representative approaches against the criteria. Analysis of the discussion highlights some open issues for future research. In response to one of the issues, we present an approach for prioritizing components for testing to maximize confidence given a limited number of test cases to be executed. Some initial results are reported in the paper.

Keywords: Web applications, model-based testing, testing prioritization, Web verification.

1 Introduction

Web applications are becoming more complex. As more and more services and information are made available over the Internet and intranets, Web sites have become extraordinarily complex, while their correctness is often crucial to the success of businesses and organizations. Although traditional software testing is already a notoriously hard, time-consuming and expensive process, Web-site testing presents even greater challenges. Complexity arises due to several factors, such as a larger number of hyperlinks, more complex interaction, frequently changing Web pages, and increased use of distributed servers. Moreover, the environment of Web applications is more complex than that of typical monolithic or client-server applications – Web applications interact with many components, such as CGI scripts, browsers, backend databases, proxy servers, etc., which may increase the risk of interoperability issues. Furthermore, many Web applications have a large number of users with no training

* Corresponding author.

on how to use the application – they are likely to exercise it in unpredictable ways. Therefore, Web sites that are critical to business operations of an organization should be tested thoroughly and frequently [9].

Modeling helps to manage the complexity of these systems. Several papers in the literature have studied the problem of web applications modeling for the sake of managing the overall development complexity. Modeling support is essential to provide an abstract view of the application. It can help designers during the design phases by formally defining the requirements, providing multiple levels of detail as well as providing support for testing prior to implementation. Support from modeling can also be used in later phases to support verification. Different models have been proposed, while others have been adapted from existing modeling techniques for other types of software [1][2][3][4][5][6][8][22][23][24][25][26][27][28][29][30][31][32][33].

In this paper we focus on Web applications testing and verification and study the different model-based approaches for managing associated complexity. In the domain of model-based testing, it is generally understood that the model is an abstraction or simplification of the behavior of the application to be tested. The model is captured in a machine readable format with the sole purpose of acting as both test sequence (trace) generator and oracle. There are many approaches to proposing a model for the purpose of Web application verification and testing. This paper studies some models that are currently applied in the field of verification and testing of web applications. Our literature survey revealed that some approaches focuses on testing the navigational aspects of web applications. Others concentrate on solving problems arising from user interaction with the browser in a way that affects the underlying process. Others are interested in dealing with static and dynamic behavior. In our bid to carry out a critical survey of the literature on using models for testing and verification of Web applications, we discovered that a common ground for classifying and comparing existing approaches is not available. This motivated our research to come up with a set of attributes serve as criteria for classifying and comparing various modeling approaches to Web application testing and verification. This set of attributes is presented in Section 2.

The analysis of a number of representative approaches against the criteria highlights some open issues for future research as discussed later. An issue of interest in this paper is that a typical Web application consists of a large number of components (i.e., front-end pages and backend processing). A Web page can be static—where content is constant for all users—or dynamic—where content changes with user input. A typical Web application could also be distributed. Accordingly, even regression testing could take weeks to test all of the test cases from a previous version [13]. Due to time and resources constraints, it would be desirable to help the tester prioritize the test cases in a way that maximize confidence given a limited number of test cases to be executed. However, the problem of prioritizing Web application components for testing did not catch enough researchers' attention. In this paper we propose an approach for an approach for prioritizing components to be tested. Such prioritization could then be used to prioritize corresponding test cases.

The rest of paper is organized as follows: Section 2 gives the comparison and categorization criteria. Section 3 discusses different approaches found in the literature in light of the criteria. Section 4 presents an approach for suggesting a prioritization as which component to be tested first. Finally we conclude and highlight some possible future work in Section 5.

2 Comparison and Categorization Criteria

System modeling is a new emerging technology. System models are created to capture different aspects of the system behavior. Several modeling languages have been developed to model state-based software systems, e.g., State Charts, Extended Finite State Machine (EFSM) [14], and Specification Description Language (SDL) [15]. System modeling is very popular for modeling state-based systems, e.g., computer communications systems, industrial control systems, etc. System models are used in the development process, e.g., in partial code generation, or in the testing process to design test cases. Over the years, several model-based test generation [14][16][17] and test suite reduction [18] techniques have been developed.

Modeling can be viewed from three different perspectives: the objective problem (security, testing etc.), the particular problem at hand (a specific case with its own characteristics e.g., ecommerce application), and finally the model type (e.g. FSM, SDL, etc.). There is still much uncertainty as to which model-based approach suits which type of Web application testing and/or verification effort. Assessing a model-based approach, in our own view, should not only be based on the underlying model expressiveness, but also on characteristics of the overall approach. We address this type of uncertainty by proposing a set of attributes to allow for classification and comparison of approaches. These assessment attributes offer more, beyond their usefulness in carrying out comparison of approaches. They can also serve as guidance to researchers attempting to develop model-based Web application testing and verification approaches. We discuss these attributes in the sequel.

Aspects Coverage: This attribute considers the Web application aspects that are being modeled by the models. These aspects are classified into three categories namely, static, dynamic and interaction aspects.

Static aspects: Static aspects of web applications include static HTML pages and the hyper links that connect the static pages with other static HTML pages. When the user clicks on a static link, a request is sent to the server to retrieve the target page.

Dynamic aspects: These aspects of web application include dynamic HTML pages that contain dynamic content and links. Dynamic contents and links are generated by backend processing based on inputs obtained from users or other supporting software.

Interaction aspects: These aspects take into consideration the user interaction with the web application. User interactions may include back page, switching to another page by typing the URL in the browser, opening multiple pages at the same time. Models can capture these types of user interactions and represent the effect on the content, behavior or the navigation.

Underlying Model: Web applications components are represented using different conceptual models, for example, some uses object relation diagram others use finite state machines model.

Perspective of Modeling: Web application models can be analyzed from different perspectives, like navigation, and behavior. These perspectives can be static or dynamic.

Objectives of the Model: Web application models have different objectives, some models objective is testing, other models objectives are implementation or design verification and model verification against a set of properties.

Source Code Requirement: Verification or testing can be a white box or a black box testing or verification. If white box testing is used by a model then the source code is required while, the black box testing requires test cases only.

Tool Support: Some models are supported by tools for automatic model generation, verification or testing, while other models are still not supported.

Expressiveness: Some models represent and convey structural, behavioral and functional aspects of web applications components for both external and internal view of the component more effectively in this case the expressiveness would be high, while other models may represent only the structural aspect or the behavioral aspect. Some models represent the external relations between components only.

Complexity: This attribute determines the complexity of the models, some models needs complex model to represent the components in term of the size and the attribute needed to represent entities and relations.

3 Critical Survey

In this section, we present a summary discussion of some representative works based on our set of attributes. The list of considered approaches in our study is not exhaustive, but we gave attention to those works we considered representative with regard to the subject under discussion. We also discuss the shortcomings associated with the different approaches considered. It is worth noting here that we used subjective ratings in evaluating the different approaches, e.g., high expressiveness and low complexity. Future work will investigate applying more quantitative objective ratings.

3.1 Model Checking-Based Verification of Web Application

Miao et al. [1] focus on automated verification of Web applications by using model checking method. The approach involves two models, the design model and the implementation model of a Web application. To verify if an implemented Web application performs in accordance with its design, the approach analyzes the design model to generate properties in temporal logic formulas that are model checked on the implementation model. Their work focuses on black-box automated verification of a Web application by using model checking method. The approach involves two

formal models: a design model denoted by *WAD*, from which the temporal logic properties for a Web application are derived, and an implementation model, denoted by *WAI*, which is model checked in order to verify those derived properties. An Object Relation Diagram (ORD) is employed to represent the design structure of a Web application, i.e., design model. Aiming at the verification of the external behavior of a Web application from client's point of view, *WAD* is intended to describe Web pages, software components interacting directly with the Web pages, and their relationships. The Kripke structure used for model checking is employed to model the implementation of a Web application, it is a type of state transition graph consisting of nodes representing the reachable states of the system and edges representing the state transitions of the system. All properties generated from *WAD* are model checked on *WAI* by using model checker *SMV* (*Symbolic Model Verifier*). *SMV* will provide a diagnostic sequence in the stack whenever a violation of the property is detected.

With regard to the tool support, this approach offers a prototype which automatically analyzes the design model to build the properties in CTL and delegates the task of property verification to the existing model checker *SMV* where the implementation model is typed in manually.

The model's level of expressiveness is considered to be moderate. While it provides a way to describe the components and the relation between them and the external view of the model very effectively, the model does not describe the low-level details and the internal behavior of each component.

The approach is considered to be of moderate complexity; the directed graph describes the external relation between components.

3.2 Testing Web Applications by Modeling with FSMs

In this approach the authors build hierarchies of Finite State Machines (FSMs) that model subsystems of the web applications [2]. This approach proceeds in two phases. Phase 1 builds a model of the web application. This is done in four steps: (1) the web application is partitioned into clusters, (2) logical web pages are defined, (3) FSMs are built for each cluster, and (4) an Application FSM is built to represent the entire web application. Phase 2 then generates tests from the model defined in Phase 1.

Tool support: They developed a research prototype in Java. It has a graphical editor to input the FSMs and the constraint descriptions. It also generates expected outputs in the form of the next state (LWP) to serve as a simple test oracle. Path generation includes edge coverage and roundtrip. Input selection is based on using an input value database. The resulting sequences of test inputs are made executable by transforming them into an *Evalid* script.

With regard to the level of expressiveness, it is high in the lowest level and low in the highest level of the hierarchy. The low level details of operations and interconnection can be observed and described; at the higher level in the hierarchy, however, the model becomes more abstract, and some of details become invisible.

The approach is considered to be of high complexity in the lowest level and low complexity in the highest level of the hierarchy. At the low level of the hierarchy, details of operations and interconnection are modeled by FSM which require many

and complex interactions but in the higher level in the hierarchy the model becomes more abstract and simpler.

3.3 An Object-Oriented Web Test Model for Testing Web Applications

Kung et al. in [3] propose a model that extends traditional test models, such as control flow graph, data flow graph, and finite state machines to web applications for capturing their test-related artifacts. Based on the proposed test model, test cases for validating web applications can be derived automatically. In this methodology, both static and dynamic test artifacts of a web application are extracted to create a Web Test Model (WTM) instance model. Through the instance model, structural and behavioral test cases can be derived systematically to benefit test processes. Test artifacts are represented in the WTM from three perspectives: the object, the behavior, and the structure.

From the object perspective, entities of a web application are represented using object relation diagram (ORD) in terms of objects and inter-dependent relationships.

In particular, an $ORD = (V, L, E)$ is a directed graph, where V is a set of nodes representing the objects, L is a set of labels representing the relationship types, and $(E \subseteq V \times V \times L)$ is a set of edges representing the relations between the objects, There are three types of objects in WTM: client pages, server pages, and components, to accommodate the new features of web applications, new relationship types are introduced in addition to those in the object-oriented programs. The new relationship types, navigation, request, response, and redirect are used to model the navigation, HTTP request/ response, and redirect relations introduced by web applications, respectively. Thus, in the ORD, the set of labels $L = I, Ag, As, N, Req, Rs, Rd$, where I : inheritance, Ag : Aggregation, As : association.

From the behavior perspective, a page navigation diagram (PND) is used to depict the navigation behavior of a web application. The PND is a finite state machine (FSM). Each state of the FSM represents a client page. The transition between the states represents the hyperlink and is labeled by the URL of the hyperlink. The PND of a web application can be constructed from an ORD. To deal with the dynamic navigation (the construction of client pages can be dynamic at runtime based on the data submitted along with the HTTP requests or the internal states of the application. Hence, the same navigation hyperlink may lead to different client pages). To model this behavior a guard condition enclosed in brackets is imposed on the transition in the PND. The guard condition specifies the conditions of the submitted data or internal system states that must be true in order to fire the transition. To detect the errors related to navigation behavior a navigation test tree is employed. A navigation test tree is a spanning tree constructed from a PND, by analyzing the tree; they can check some properties, such as reachability and deadlock, of the navigation behavior. At the same time, a set of object state diagrams (OSDs) are used to describe the state behavior of interacting objects. It can represent the state-dependent behavior of an object in a web application. The state-dependent behavior for an aggregate object then can be modeled by a composite OSD (COSD) of the corresponding OSDs.

The structure perspective of the WTM is to extract both control flow and data flow information of a Web application. To capture control flow and data flow information, the Block Branch Diagram (BBD) and Function Cluster Diagrams (FCD) are

employed in the WTM. The BBD is similar to a control flow graph. It is constructed for each individual function of a Web application to describe the control and data flow information, including the internal control structure, variables used/defined, parameter list, and functions invoked, of a function. Therefore, the BBD can be used for traditional structural testing of each individual function; the FCD is a set of function clusters within an object. Each function cluster is a graph $G = (V, E)$, where V is a set of nodes representing the individual functions and $E \subseteq V \times V$, is a set of edges representing the calling relations between the nodes.

The approach offers a very high level of expressiveness. Different models are used to describe external, behavioral and internal aspects of components which can express the model effectively.

The approach is considered to be of very high complexity. Many models are used to describe the internal, behavioral and external structure of components so the overall system model is very complex.

3.4 Formal Verification of Web Applications Modeled by Communicating Automata

Haydar et al. in [4] devise an algorithm to convert the observed behavior, which they called a browsing session, into an automata based model. In case of applications with frames and multiple windows that exhibit concurrent behavior, the browsing session is partitioned into local browsing sessions, each corresponding to the frame/window/frameset entities in the application under test. These local sessions are then converted into communicating automata. They did an implementation for a framework which includes the following steps: The user defines some desired attributes through a graphical user interface prior to the analysis process. For example, reachability properties, and the checking for frame errors, frames having same name are not active simultaneously. These attributes are used in formulating the properties to verify on the application. A monitoring tool intercepts HTTP requests and responses during the navigation of the Web Application Under Test (WAUT). The intercepted data are fed to an analysis tool, which continuously analyzes the data in real time (online mode), incrementally builds an internal data structure of the automata model of the browsing session, and translates it into XML-Promela. The XML-Promela file is then imported into aSpin, an extension of the Spin model checker. ASpin then verifies the model against the properties, furthermore the model checking results include counterexamples that facilitate error tracking.

The approach is supported with a framework that is composed of; GUI to collect desirable properties from the user, network monitoring tool to intercept HTTP request and response, analysis tool that builds the communicating automata based on the received data. The model is fed into aSpin for verification.

The approach offers a low level of expressiveness, as the model describes a session or multiple sessions, which may not give a full description of the complete model of the system; it depends on how the user will interact with the application.

The approach is considered to be of high complexity; based on the user input the FSM can get complex.

3.5 Verifying Interactive Web Programs

Licata et al. in [5] describe a model checker designed to identify errors in web software. A technique for automatically generating novel models of web programs from their source code was presented. These models include the additional control flow enabled by user operations. They presented a powerful base property language that permits specification of useful web properties, along with several property idioms that simplify specification of the most common web properties. The authors model a web program P by its web control-flow graph (WebCFG). The WebCFG is an augmented control-flow graph (CFG). User interaction control flows are being added to the model to build a sound verification tool. The authors reduce user operations to primitive user operations proposed by Graunke et al. [8]. All traditional browser operations can be expressed in this calculus; they just account for switch and submit. Then they construct the WebCFG completely automatically from the source of a web program using a standard CFG construction technique followed by a simple graph traversal to add the post-web-interaction nodes and the web-interaction edges. The resulting model and properties are checkable by language containment. This work doesn't address the concurrency issues resulting from multiple simultaneous accesses to a server by different clients.

With regard to tool support, the authors implemented their own model checker tool to support their approach.

The approach models are meant to prove properties of interactive web sites by discovering user operation- related bugs, as well as providing a method for verifying all-paths properties of interactive web sites.

The approach offers high level of expressiveness. CFGGraph describe details of behaviors of components and how these interact with each other. In addition, adding the user operations to the model makes the model describe the behavioral aspect based on the user operations.

The approach is considered to of very high complexity; CFGGraph is very complex, especially when the user operation is involved in the model.

3.6 Web Site Analysis: Structure and Evolution

Ricca et al. in [6] adapts an approach to analyze, test, and restructure web application based on a reverse engineering paradigm. They didn't propose models and formalisms to support the design of web applications; instead, based on the assumption that a web application already exists, they investigate different well established methods for the analysis, testing and re-structuring of traditional software systems, adapting them to the case of Web applications. In [6] web application is modeled as a graph; nodes and edges are split into different subsets. Nodes subsets are a set of all web pages; a set of frames for one web page; and a set of all frames.

Edges are also split into three subsets according to the kind of target node; a set of hyperlinks between pages or a relation showing the composition of web page into frames; a set of the relations between frames and pages; as they show which page in which frame is loaded; and a set of relations showing the loading of a page into a particular frame. The name of the frame is given as a label next to the link. This model is implemented in ReWeb. The ReWeb [7] tool consists of three modules: a

Spider, an Analyzer and a Viewer. The Spider downloads all pages of a target web site, starting from a given URL and providing the input required by dynamic pages, and then it builds a model of the downloaded site. The Analyzer uses the UML model of the web site and the downloaded pages to perform several analyses. Since the structure of a Web application can be modeled with a graph, several known analysis, working on graphs, such as flow analysis and traversal algorithms can be applied. The Viewer provides a Graphical User Interface (GUI) to display the Web application view as well as the textual output (reports) of the analyses.

With regard to supportability, the approach is supported by the ReWeb tool. The ReWeb tool can periodically download the entire set of pages in a site. Results of the analyses are then provided to the user, by exploiting different visualization techniques. Colors are employed in the history view, while structural and system views are enriched with powerful navigation facilities. Pop-up windows associated to nodes are used to show the textual results of the structural analyses.

The level of expressiveness is low; the model described by directed graph only. The approach is considered to be of low complexity; only a directed graph is involved in the model.

3.7 Summary

Table 1 shows the summary of the 6 different methods described.

Table 1. Summary of Findings

Method	Aspect type	Model	Perspective	Objective	Source code	Tool Support	Expressiveness	Complexity
Miao et al.	Static + dynamic	ORD	Navigation + behavior	Implementation verification against design Testing	Yes	Prototype	Moderate	Moderate
Andrwes et al.	Static + dynamic	FSM, AFSM	Navigation + behavior	Testing	No	Prototype	Low	Low
Kung et al.	Static + dynamic	ORD, PND, OSD, BBD, FCD	Behavior + navigation	Testing	Yes	None	Very High	Very High
Haydar et al.	Static + dynamic	Communicating automata	Navigation + behavior	Model verification against defined properties	No	GUI + network monitoring tool + analysis tool	Low	High
Licata et al.	Interaction	WebCFG	Interaction behavior	Model verification against interactive properties	Yes	Implement a model checker	High	Very High
Ricca et al.	Static	Directed graph	Navigation	Original design verification during evolution and Testing	Yes	ReWeb	Low	Low

4 Components Testing Prioritization

From Table 1 we can see that methods discussed are lacking ways to prioritize Web application components for testing. This untreated aspect is very important especially when we know that development and deployment cycles of Web applications are dramatically becoming short, and testing is often considered a cost-intensive and time-consuming process. Here, we give several suggestions which could be investigated more thoroughly in future works. First solution is to apply an algorithm

to find the minimum independent dominating set on the graph based model, then we can consider these set as the highest priority components to test. The rationale here is that these dominating components can be regarded as super components because they are connected to many other components. Also the components in this way are either dominating or dominated by others; so, all components that may lead to other components can be tested. Another suggestion is to rank components based on the degree value of a node. So, an important node is involved in a large number of interactions. For directed networks, there are two notions of degree centrality: one based on fan in-degree and the other on fan out-degree. A node with high fan in-degree is ranked higher than those of less degree; since high fan in-degree means that most probably many components will leads to this component. *Betweenness* measure can be used to rank components. The measure reflects the intuition that an important node will lie on a high proportion of paths between other nodes in the network.

In order to see how these suggested methods works, we will apply these methods on an ORD model design (Fig. 1).

4.1 Minimum Independent Dominating Set Method (MIDSM)

A dominating set D of a graph $G(V, E)$ is a subset of V in which each vertex $v \in (V - D)$ is adjacent to at least one vertex $u \in D$, i.e., $(v, u) \in E$. An independent dominating set is a dominating set (where D is independent, i.e., $(u, v) \notin E$, for all $u, v \in D$). Since finding the minimum independent dominating set is NP-Complete problem [21], we will use a greedy algorithm to find a set that is as minimum as possible. First, we will find the minimum independent dominating set by using a greedy algorithm which can be applied on undirected graph and it will choose a node with maximum degree and delete the neighbors. So, the first step is to convert the model to an undirected graph, the result can be seen in Fig. 2.

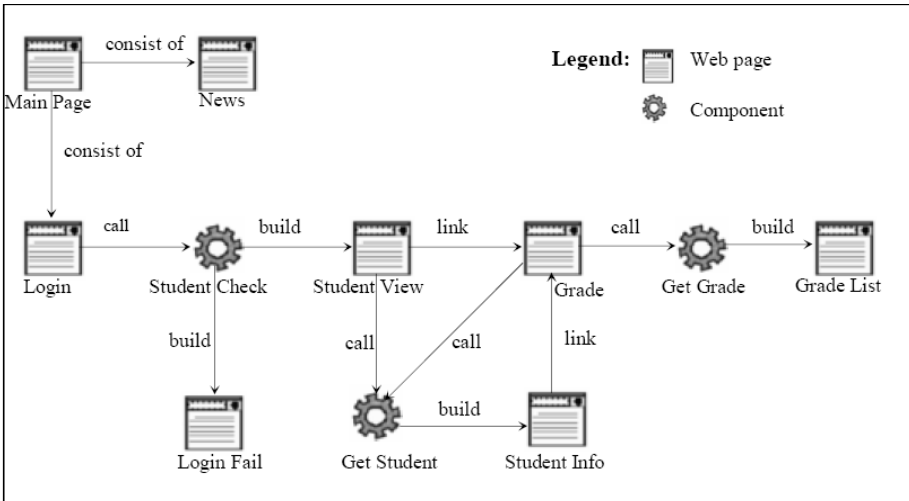


Fig. 1. An ORD design model [1]

If we apply the greedy algorithm we will choose node H which corresponds to the grade web page as the first node because it has degree of four which is the maximum and delete all neighbors. Now we can select either node A or node D since they have the highest degree which is two, let us select A assuming there is no any other criteria for selection. Now we can select node D and then select node K. So, the final set is H, A, D and K.

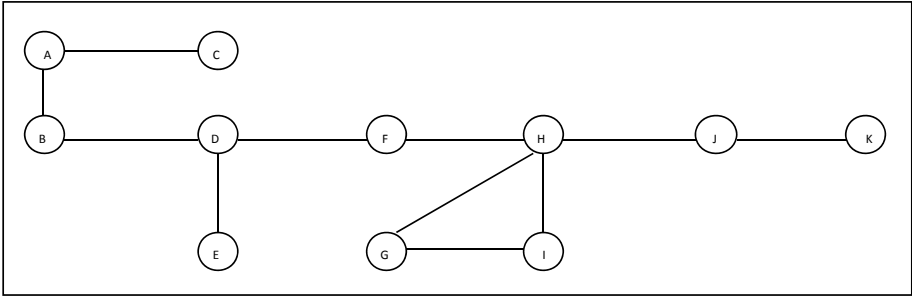


Fig. 2. The undirected graph of the ORD model

Analysis

The grade web page is used and uses more components than the other nodes so it is indeed an important page. The main page was selected as an important page but it is not since it only contains links to two pages so it is static. Student check component is important since it check for the validity of the user. The grade list web page is selected as an important page but it is not since it only contains the final results which depend on the get grade component which is more important. In addition, the method missed by two pages which is more than that of the other components. The weakness of this approach is when there is more than one node with the same degree; in this case, which one to select? We could define more criteria for selection like the type of the node and the type of the edge which can impact the selection. Another weakness is not considering the importance of the direction which may impact the importance of the components. Also, if we delete the neighbors, we might actually delete an important component or page.

4.2 The Degree Measure Method (DMM)

The idea behind using a degree measure of importance in a network is the following: An important node is involved in a large number of interactions. Formally, for an undirected graph G , the degree centrality of a node $u \in V(G)$ is given by $Dm(u) = \deg(u)$ [19]. For directed networks, there are two notions of degree measure: one based on fan in-degree and the other on fan out-degree, we will use the fan in-degree measure. Now let us rank the components based on the fan in-degree. Get student component and grade page have degree 2 which is the highest degree. Then news, login, login fail, student view, student info, grade list pages, and student check component with degree of one. The main page has lowest degree with degree of zero.

Analysis

The result show better ranking of importance because if the components which have high fan in-degree fail then many other components will fail to get the services. Get student and grades page are used by more components than the other components, so any failure in these components will make the other component fail. The issue is that we might have many components with same degree, the question is how we can prioritize these with same degree; we might add more criteria like the component type and the fan in-edges types.

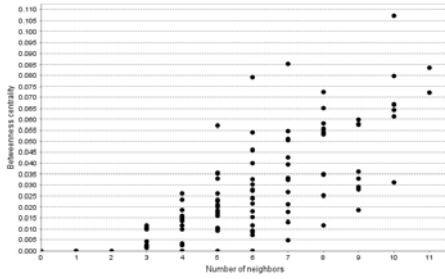
4.3 Betweenness Measure Method (BMM)

Now let us use the *betweenness* measure to rank the importance of the components. The idea behind this measure is the following: An important node will lie on a high proportion of paths between other nodes in the network. Formally, for distinct nodes, $u, v, w \in V(G)$, let σ_{uv} be the total number of shortest paths between u and v and $\sigma_{uv}(w)$ be the number of shortest paths from u to v that pass through w . Also, for $w \in V(G)$, let $V(u)$ denote the set of all ordered pairs, (u, v) in $V(G) \times V(G)$ such that u, v, w are all distinct. Then, the betweenness measure of w , $B_m(w)$, is given by $B_m(w) = \sum_{(u,v) \in V(w)} \frac{\sigma_{uv}(w)}{\sigma_{uv}}$ [20]. First, all shortest paths between any pairs of components in the model are found. Then we will go over all components and see on which paths they exist. The main page and the news page do not come between any other components in a path so their B_m is 0. Login exists on 8 paths so its B_m is 8. Student check comes between 14 components on different shortest paths so its B_m is 14. Student view's B_m is 15. Get student comes between 4 components on different shortest paths so its B_m is 4. Student info page's B_m is 3. Grade component's B_m is 13. Get grade component exists on 7 paths so its B_m is 7. The grade list page is not between any other pages so its B_m is 0. From the results we can see that student view page has the highest B_m then student check component and then the grade page, after that login, get grade, and get grade and student info page.

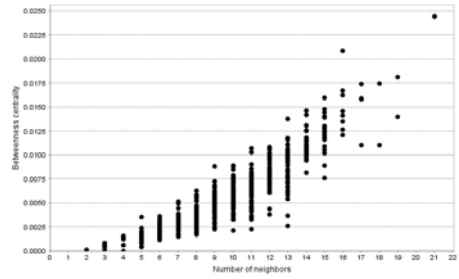
Analysis

The results show good ranking because if any components which comes between many other components fail, then the other components will fail to reach the other component, which means those components with high B_m are bottle nicks so they are important and their priority in testing should be high. The weakness in this approach is that we might have components with the same B_m s, the question is which components is more important within these components, so we need to add more attributes like the type of components, and the type of edges in these components.

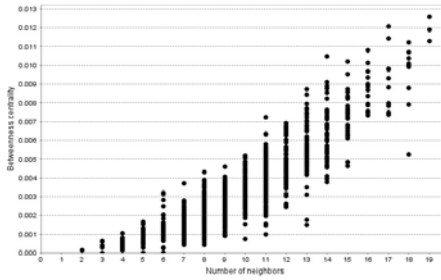
We conducted very rough set of experiments on different randomly developed networks of components to reflect applications of different sizes. The sizes of the networks were chosen to be between 100, 500, and 1000 nodes. The topologies of the networks were chosen randomly. Prioritizations of the components are plotted in Fig 3 below.



A Web Application with 100 components



A Web Application with 500 components



A Web Application with 1000 components

Fig. 3. Betweenness Measurements of Arbitrary Web Applications

Fig. 3 shows that components with more neighbors are of higher betweenness measurements; and hence are of higher testing priority.

5 Conclusion and Future Work

In this paper we proposed set of attributes for classifying and comparing model-based Web applications testing and verification approaches. We discussed six different representative analysis models that are currently applied in the field. We summarized our discussion in Table 1 which reveals that the methods discussed are lacking ways to prioritize web application components for testing. We suggested three methods to allow for prioritizing components: MIDSM, DMM, and BMM. We illustrated the suggested methods on an ORD design model. The results show that the MIDSM has some shortcomings and may miss important components and consider not important components. The DMM and BMM show better results, with some issues. The issues can be addressed by incorporating more attributes and criteria for selections like the type of the components, and the edges, these attributes in addition to others can be investigated in future work. Also, we plan to investigate combining the DMM and BMM together to rank the components by assigning a percentage for each measure in future work as well. The percentage can be learned from experience, and using machine learning methods to find the best percentage. It is worth noting here though

that in this paper we only demonstrated the approach using an illustrative example; in future work, we will conduct more rigorous analysis of the different methods.

Another task for future work will focus on replacing the subjective scheme we used for rating approaches (e.g., with regard to complexity and expressiveness) with more qualitative one.

Acknowledgements. The authors wish to acknowledge King Fahd University of Petroleum and Minerals (KFUPM) for utilizing the various facilities in carrying out this research.

References

- [1] Miao, H., Zeng, H.: Model Checking-based Verification of Web Application. In: Proceedings of 12th IEEE International Conference on Engineering Complex Computer Systems pp. 47–55 (2007)
- [2] Andrews, A., Offutt, J., Alexander, R.: Testing Web Applications by Modeling with FSMs. *Software Systems and Modeling* 4(3), 326–345 (2005)
- [3] Kung, D.C., Liu, C.H., Hsia, P.: An Object-Oriented Web Test Model for Testing Web Applications. In: Proceedings of the 1st Asia-Pacific Conference on Web Applications, pp. 111–120. IEEE Press, New York (2000)
- [4] Haydar, M., Petrenko, A., Sahraoui, H.: Formal Verification of Web Applications Modeled by Communicating Automata. In: Proceedings of the 24th IFIP International Conference on Formal Techniques for Networked and Distributed Systems, Madrid, Spain, pp. 115–132 (2004)
- [5] Licata, D.R., Krishnamurthi, S.: Verifying interactive web programs. In: Proceedings of the IEEE International Conference on Automated Software Engineering, pp. 164–173. IEEE Computer Society, Los Alamitos (2004)
- [6] Ricca, F., Tonella, P.: Web site analysis: Structure and evolution. In: Proceedings of the International Conference on Software Maintenance, pp. 76–86 (2000)
- [7] Ricca, F., Tonella, P.: Building a Tool for the Analysis and Testing of Web Applications: Problems and Solutions. In: Margaria, T., Yi, W. (eds.) TACAS 2001. LNCS, vol. 2031, pp. 373–388. Springer, Heidelberg (2001)
- [8] Graunke, P.T., Findler, R.B., Adsul, B., Felleisen, M.: Modeling Web Interactions. In: Degano, P. (ed.) ESOP 2003. LNCS, vol. 2618, pp. 238–252. Springer, Heidelberg (2003)
- [9] Benedikt, M., Freire, J., Godefroid, P.: VeriWeb: Automatically Testing Dynamic Web Sites. In: Proceedings of 11th International World Wide Web Conference (2002)
- [10] Sampath, S., Bryce, R., Viswanath, G., Kandimalla, V., Koru, A.G.: Prioritizing User-Session-Based Test Cases for Web Application Testing. In: Proceedings of IEEE Int. Conf. Software Testing, Verification, and Validation, pp. 141–150 (2008)
- [11] Bryce, R.C., Sampath, S., Memon, A.M.: Developing a Single Model and Test Prioritization Strategies for Event-Driven Software. *IEEE Transactions On Software Engineering* 37, 48–64 (2011)

- [12] Korel, B., Tahat, L.H., Harman, M.: Test Prioritization Using System Models. In: Proceedings of the 21st IEEE International Conference on Software Maintenance (2005)
- [13] Rothermel, G., Untch, R.H., Chu, C., Harrold, M.J.: Prioritizing Test Cases for Regression Testing. *IEEE Trans. Software Eng.* 27(10), 929–948 (2001)
- [14] Cheng, K., Krishnakumar, A.: Automatic Functional Test Generation Using The Extended Finite State Machine Model. In: Proceedings of ACM/IEEE Design Automation Conf. pp. 86–91 (1993)
- [15] Dssouli, R., Saleh, K., Aboulhamid, E., En-Nouaary, A., Bourhfir, C.: Test Development For Communication Protocols: Towards Automation. *Computer Networks* 31, 1835–1872 (1999)
- [16] Dick, J., Faivre, A.: Automating the Generation and Sequencing of Test Case from Model-Based Specification. In: Proceedings of International Symposium on Formal Methods, pp. 268–284 (1992)
- [17] Vaysburg, B., Tahat, L., Korel, B.: Dependence Analysis in Reduction of Requirement Based Test Suites. In: Proceedings of ACM International Symposium on Software Testing and Analysis, pp. 107–111 (2002)
- [18] Korel, B., Tahat, L., Vaysburg, B.: Model Based Regression Test Reduction Using Dependence Analysis. In: Proceeding of IEEE International Conf. on Software Maintenance, pp. 214–223 (2002)
- [19] Nieminen, J.: On centrality in a graph. *Scandinavian Journal of Psychology* 15, 322–336 (1974)
- [20] Freeman, C.: A set of measures of centrality based on betweenness. *Sociometry* 40, 35–41 (1977)
- [21] Garey, M.R., Johnson, D.S.: Computers and intractability. A guide to the theory of NP-completeness. W. H. Freeman, San Francisco (1979)
- [22] Conallen, J.: Modeling web application architectures with UML. *Communications of the ACM* 42(10), 63–71 (1999)
- [23] de Alfaro, L.: Model checking the world wide web. In: Berry, G., Comon, H., Finkel, A. (eds.) CAV 2001. LNCS, vol. 2102, pp. 337–349. Springer, Heidelberg (2001)
- [24] Alpuente, M., Ballis, D., Falaschi, M.: A rewriting-based framework for web sites verification. *Electr. Notes Theor. Comput. Sci.* 124(1), 41–61 (2005)
- [25] Chen, J., Zhao, X.: Formal models for web navigations with session control and browser cache. In: Davies, J., Schulte, W., Barnett, M. (eds.) ICFEM 2004. LNCS, vol. 3308, pp. 46–60. Springer, Heidelberg (2004)
- [26] Bordbar, B., Anastakis, K.: MDA and Analysis of Web Applications. In: Draheim, D., Weber, G. (eds.) TEAA 2005. LNCS, vol. 3888, pp. 44–55. Springer, Heidelberg (2006)
- [27] Winckler, M., Palanque, P.: StateWebCharts: A formal description technique dedicated to navigation modelling of web applications. In: Jorge, J.A., Jardim Nunes, N., Falcão e Cunha, J. (eds.) DSV-IS 2003. LNCS, vol. 2844, pp. 61–76. Springer, Heidelberg (2003)
- [28] Han, M., Hofmeister, C.: Modeling and verification of adaptive navigation in web applications. In: ICWE. pp. 329–336 (2006)
- [29] Di Sciascio, E., Donini, F., Mongiello, M., Piscitelli, G.: Web applications design and maintenance using symbolic model checking. In: Proceedings of the European Conference on Software Maintenance and Reengineering, pp. 63–72. IEEE Computer Society, Los Alamitos, CA, USA (2003)

- [30] Castelluccia, D., Mongiello, M., Ruta, M., Totaro, R.: Waver: A model checking-based tool to verify web application design. *Electr. Notes Theor. Comput. Sci.* 157(1), 61–76 (2006)
- [31] Bellettini, C., Marchetto, A., Trentini, A.: Webuml: reverse engineering of web applications. In: *SAC*, pp. 1662–1669 (2004)
- [32] Wu, Y., Outt, J.: Modeling and testing web-based applications. Technical report, George Mason University (2002)
- [33] Syriani, J.A., Mansour, N.: Modeling Web Systems Using SDL. In: Yazıcı, A., Şener, C. (eds.) *ISCIS 2003. LNCS*, vol. 2869, pp. 1019–1026. Springer, Heidelberg (2003)

Web Interactive Multimedia Technology: State of the Art

Asma Md Ali and Joan Richardson

School of Business Information Technology and Logistics,
RMIT University, VIC 3001, Melbourne
{Asma.mdali,Joan.richardson}@rmit.edu.au

Abstract. Elluminate a web interactive multimedia technology (WIMT), which is an information and communication system, was introduced in a large metropolitan University. Its attributes are outlined in this paper, from just text-based to more complex-based features. The system incorporates several multimedia features, such as chat, audio, video, polling, whiteboard and desktop sharing. This system provides real time collaboration. When used in a university teaching and learning environment, it enables immediate feedback between participants across physical space. This gives an added opportunity for interactivity in an online learning environment. Relationship building capacity between academic and student is a vital component of learning. WIMT enables augment learning through interaction between academic and student.

Keywords: Web interactive multimedia technology, Blended environment.

1 Introduction

This short paper introduces web interactive multimedia technology (WIMT), an information and communication system used by the university. Information and communication technology (ICT) continues to shape public and professional interactions. With the emergence of the internet and web technology, information is ready-made and data easily accessible. Hence, accessing and disseminating information becomes even easier to users but challenging for developers in the web development lifecycle. The Internet-based World Wide Web has had an enormous impact on web applications and society due to features that provide a means for collaborative learning, open access to information and social networking.

Universities began to adopt blended learning approaches to teaching which enriched the learning experience for all students irrespective of age and nurtured life-long learning. Blended learning incorporates learners' interactions with lecturers, online learning via interactive multimedia systems and self-study.

This research is in progress at a large metropolitan university with several campuses in Southeast Asia that have trialed Elluminate. University wide implementation is planned.

The next section will explain the research motivation in looking at the extended opportunity for interactivity in an online environment using web technology.

2 Motivation

Web technology has been receiving the attention of IT professionals since the development of the internet. It is similar to other human made technologies such as the telephone and television in that it is a tool used to disseminate information. Humans are then responsible for evaluating and comprehending the message and ascertaining its usefulness in their own particular context. Users are demanding improvements in computer based message delivery environments, modes of delivery and message composition. Designers need to keep pace with continuously changing available technologies. The stakeholders involved include: builders, designers, content developers, web maintenance roles and the user (in specific purpose and context).

Although there is no single definition for the term 'web technology', there are well-known characteristics of the web that researchers and practitioners agree need to be considered to design, develop, and implement web delivery systems in the higher education context. These characteristics include: ubiquity, existence of open standards, interlinking, easy access to information and services and easy content creation [1], [2].

The web can be categorized into fixed and/or mobile access systems according to the devices and applications used. The fixed web is where the end-user utilises wired devices like a desktop to access the internet whilst the mobile web is where the end-user utilises mobile devices to access the internet, such as iPods, notebooks and mobile phones. Wireless devices are beginning to be implemented in universities to facilitate a collaborative learning environment [3]. This study focuses on the fixed web, specifically desktops and does not include the mobile web area.

Web technologies such as Active Server Pages (ASP), JavaScript, VisualBasic Script, Structural Query Language (SQL), Open Database Connectivity (ODBC), AJAX and streaming video technology have pushed the dynamic experience of users. Web interactive multimedia programs are being used in various fields such as business, education, training and health care. With the recent advances in artificial intelligence, knowledge representation and technologies for information systems, there are various methodologies being used in modeling and developing interactive multimedia programs [4].

However, there are few research studies being done on the deployment of systems which look at best practice. The study reported here examines case studies of effective use of WIMT particularly for university learning and teaching. It is posited that the use of interactive multimedia programs can provide a richer teaching and learning environment and enable collaborative work using the web element that would otherwise not be possible. Baharun and Porter [5], Boulay, et al., [6], Craig, et al., [7], and Cody [8] describe cases where a website has augmented the teaching of statistics; online material has been developed to teach molecular biology; web based lecture technologies have been used to teach medical students and an online database has augmented the teaching of dance.

Iivari, Isomaki and Pekkola [9] mentioned that there are signs of the Information Systems (IS) research community broadening its focus to include investigation of user-oriented design of multi-media systems and research methods. There are calls for


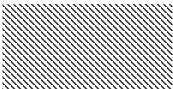


researchers to enter the web and blended learning research community [10]. According to Ivory and Megraw [4], the pattern for web research started with a back-end approach (the database and infrastructure) and then shifted towards a front-end approach (interface and users). The focus towards user-oriented methods is emerging in the IS research community [9]. Furthermore, there is a need to do this research as the quality, extent and impact on learning of ICT use in blended and online learning environments remains an under-researched area [11].

In Australia, an exploratory study on the impact of web technologies on learning in universities has commenced [12]. One large metropolitan university has completed a pilot study and is currently carrying out a university wide implementation. This is where the case study described in this paper is being conducted. Among the web technologies being researched are web-based lecture technologies (WBLT) and Web3D. There are published resources and opportunities for the researcher to develop skills and knowledge in various types of web technologies focusing on collaboration and interactivity such as Elluminate which has been adopted by some schools and universities in Australia.

3 Glossary of Terms

Some of the terms used in this paper are defined as in Table 1.

Table 1. The glossary of terms used in this paper

Term	Acronym	Definition
Blended environment		Mixed web and face-to-face interaction in a traditional classroom.
Elluminate	eLive	A web interactive multimedia technology artifact.
Information Systems	IS	A field of study that incorporate technology, societies and organizations.
Interactive		Non-static; response to user input.
Multimedia		Combination of two or more: audio, text, image, animation and video.
Web Interactive Multimedia Technology	WIM Technology; WIMT	Systems that combine more than two media, respond to user input and delivered/accessed through the internet.
Web Technology		System access through desktop from the internet infrastructure.

4 State of the Art

Universities provide web delivery of learning and teaching resources, specifically targeted at adults who need to take care of their families, manage their career and pursue a higher education. One of the key deliverables of web learning is web interactive multimedia systems that facilitate the learning and teaching process.

4.1 Blended Environment

One approach in using web technology for learning in universities is a blended environment. It is a mix of web and face-to-face interaction in a traditional classroom or lab or physical space. Positive findings using blended learning include indicating learning quality in an online community using an interaction based approach [14].

Eklund, Kay and Lynch [15] state that the growing trend towards blended learning environments recognizes that the use of ICT augments face-to-face delivery, and provides unique experiences that assist in achieving desired learning goals. Blended learning allows for learning and teaching practices to be combined into a custom made learning experience for each individual learner [15]. Blended learning has been successful because it commonly emerges as a delivery technique from a process of planning and analysis. There is also evidence to show that it is a learning design implicit in many success models. According to Zhang et al., [16] the environment places emphasis on learner-centre activity and system interactivity that can enable distance learners to outperform traditional classroom students. Therefore, blended environments have the potential to improve traditional classroom learning [16].

4.2 Text-Based

Collaborative activities where students can use e-mail, forums, bulletin boards and share and edit documents online arise as alternatives to the more rigid Learner Management Systems, like WebCT. The collaboration with students renews the teacher/learner relationship whilst maintaining immediacy and minimizing the need for technical expertise. Inter-person collaboration and knowledge building is seen as one of the most effective way for adults to learn. The future of lifelong learning depends on reducing the gap between the conceptual arguments and real effective implementations of WBLT.

4.3 Audio-Based

There is also evidence that web-based lecture technology (WBLT) is used by students as a study tool to complement face-to-face lectures [12]. Students report using WBLT to support their learning by checking over notes, reviewing difficult concepts, preparing for exams and listening to missed lectures. The acceptance of this delivery for private lecture study has been overwhelmingly positive and shifts towards being a successfully embedded technology.

Further questions of how WBLT's can be integrated into the delivery of a unit of study by adjusting the lecturing style and how a course can be delivered to make the most effective use of web-based lectures are yet to be answered [12]. The feedback

from staff and students in Gosper's [12] study also raised questions relating to changes in teaching style, good teaching practice, perspectives on the use of WBLT, the best way to support learning, different uses to support learning (rather than delivery), differences across disciplines and modes of delivery, as well as other ways to enhance learning, teaching and curriculum design. Gosper's [12] study also confirms student's appreciation of the convenience and flexibility offered by time and access to lectures.

4.4 Video-Based

With increased demands posed by work and family commitments, one way to address students need for flexibility is to provide easy access to lecture recordings. In addition to flexibility, the impact of these technologies is generally positive on students' learning [17]. McElroy and Blount's [18] surveyed 411 students who had used iLecture technology. More than 75% of students agreed that iLecture enhanced the course when compared to other subjects that did not utilise the technology [18]. Soong, Chan and Cheers [19] reported on a similar study conducted in Singapore, where video-recorded lectures had been created to support students' learning from traditional face-to-face lectures. In a survey of 1160 students, they found that 94.9% agreed that the video-recorded lectures were useful in relation to their studies. The most popular reasons for using video recorded lectures were for reviewing difficult parts of the lectures and for exam preparation [19].

4.5 Audio-Video-Based

Web-based Lecture Technologies (WBLT) have been studied by four universities in Australia namely Macquarie University, Murdoch University, Flinders University and The University of Newcastle. Lectoria, was the WBLT, researched as it has the capacity to integrate the Learning Management System (LMS), Blackboard and WebCT. Research on audio recording technology and linear video technology of live lectures was conducted using a case study approach comprised of a mixed method: survey, questionnaire and interview [12]. This study would like to look at Web Interactive Multimedia Technology that has more features than linear audio and video technology.

In Gosper's [12] report examining cases where WBLT had been used by the students and staff, 76% of students and 54% of staff rated the experience positively. However, there was inconsistency between between student and staff perceptions of the benefits of WBLT for learning (80% students compared with 49% of staff agreed) and achievement of better results (67% students compared with 30% staff agreed) [12]. An exploration of the impact of WBLT on learning and teaching is of interest to the higher education sector. This is because of the increasing demand from students for flexible access to educational opportunities, and substantial investments by institutions in this area. Relationship building between academics and students is a vital component of learning. The potential to substantially improve teaching practice, to improve the students learning experience and to contribute to the development of effective mechanisms for the identification, development, dissemination and embedding of individual and institutional 'good practice' in universities are exist.

4.6 Complex-Based WIMT

It is useful to evaluate these systems to obtain a better understanding of the effectiveness of multimedia programs. The complex-based WIMT incorporate the five multimedia elements, which are text, audio, video, graphic and animation. Elluminate, a web-based conferencing software application for real time collaboration, provides opportunities to conduct tele-tutorials in a virtual classroom setting. Elluminate has been used as the case study described in this paper. It enables instructors and users to have real-time discussions while viewing PowerPoint slides, web sites, whiteboard and shared applications - all of which are interactive. It also offers text messaging capabilities, ad-hoc surveys (polling) and basic assessments. Classes can be recorded for later playback.

This research looks at how Elluminate functionality and features have been integrated into the teaching and learning activities. What have been the pragmatic learning and teaching activity design and technical issues? What features have been used? What has been the impact of the new technology on learning and teaching resources and delivery modes?

5 The Problems

The university IT services decided to do a pilot study in order to implement a complex-based web interactive multimedia technology in the university. However, the technological infrastructure has delayed the committee to make decision on implementing it. Another problem is lack of staff training during pilot as only volunteers are called to participate. Surprisingly there exist staff that use the technology without involving themselves in the pilot study.

6 The Method

A qualitative method was used to enable the researcher to answer the research question: How did the university implemented web interactive multimedia technology (WIMT)? Furthermore, this study will be driven by the interpretive paradigm as this research attempts to understand a phenomenon through accessing the meanings that participants assign to them [20]. This research adopts the case study research method as this is an exploratory study and the researcher needs to obtain in-depth data on WIMT implementation in the university environment. The case study was conducted at a large metropolitan university that has conducted a pilot study of Elluminate use and is currently implementing the application. A key person from the pilot project committee, and a coordinator of a graduate program were interviewed using a semi-structured interview approach. This enabled two points of view to be collected, from the administration implementer and the academic user that used the technology to complete work tasks. Ethics approval was obtained to record the interviews and use transcripts and written notes for this research. The one-on-one interview took approximately 30-40 minutes. The implementer commented on issues related to the WIMT implementation in the large metropolitan university followed by the user perspectives.

The next section will explain the case study in looking at the extended opportunity for interactivity in an online environment when a web interactive multimedia technology is implemented in a university.

7 The Case Study

Universities and colleges present a unique setting to explore the deployment of new technologies. Some universities purport that teaching is one of the top priorities, with research and service playing important roles. However, teaching with WIMT can require more time and effort to prepare quality learning and teaching resources. University faculty members tend to have some control over what content is taught, and more control over how the content is taught and assessed [13].

Learning via the Internet, intranets and extranets is increasingly understood to be a subset of e-learning (technology supported learning). Web-based learning is an identifiable artifact of learning objectives, content and interactions. There are efforts to determine the factors that create successful web-based learning programs which include establishing a basic framework covering dimensions as diverse as the pedagogical; technological; interface design; evaluation; management; resource support; and ethical considerations. There also exist discussions of cognitivism and constructivism in learning that focus on achieving higher level learning in independent, self-reliant learners who can imply a range of strategies to construct their own knowledge.

Web based learning is seen as a means to modify or influence the behavior of clients and hence to achieve corporate goals within market. The corporate sector's recognizes the benefits of e-learning and extends these beyond their investment in their employees. For example, Melbourne water developed a website to educate children on water conservation, believing that educating users online will help increase waste-water recycling to 20 per cent by 2010.

8 The Findings

8.1 Case Study 1: Piloting Elluminate in a University - Academic Developer

Elluminate was piloted and implemented in the university after three types of software had been evaluated. Initially there were some technical issues for end-users of Elluminate caused by the ICT infrastructure. However, as the software and infrastructure have matured, the university has decided to implement Elluminate to practically realize and reap the benefits of WIMT:

“The university have been waiting for a software and the technology (including bandwidth, reliability of the technical aspect) to come closer together to make it possible for the idea of all things you can do in Elluminate being useful for people. You can use it for distance learning, tutorial, professional development and software training”.

This includes making use of the complex-based WIMT features, which are messaging, audio and video conferencing, audience response tools, whiteboard and

application sharing. A pilot study was conducted to test the technical capacity of the WIMT rather than the actual learning and teaching aspects, the core activity of a university: “The pilot study look more at the technical side of things. It should have also addressed the learning and teaching aspect”.

As the pilot committee was satisfied with the pilot study, Elluminate was to be implemented to university wide. This involved three faculties. So the next step was to create a “good communication strategy” to support the implementation. To support uptake of the recommended WIMT, successful examples of WIMT implemented in learning and teaching activities including “how they were used and what benefits were obtained from using it” were published and demonstrated for others to see the practicality and benefits. The published examples were intended to support staff in extending their learning beyond the traditional boundaries.

8.2 Case Study 2: Exploring Elluminate in a University - Academic Lecturer

During the pilot study, academics were asked to volunteer. However, the user interviewed was not directly involved in trialling the technology in the pilot study. She came across Elluminate when she was setting up a graduate neurology course for distance education students. The university informed her that the previous virtual learning environment system was no longer available and the university was currently adopting and piloting Elluminate: “I needed to have a virtual classroom connected (for my distance education students) and they said we are using Elluminate now.” She straight away installed Elluminate and found it was user friendly and easy to use. She did not go to the formal university training sessions but she managed to explore it on her own: “Although I missed out on the (formal) training, it was quite intuitive (to use it)”. When a certain task to accomplish a planned activity was a bit of a challenge, she contacted the university Elluminate support team in the teaching and learning unit and joined a network of users involved in the pilot who were exploring the features and functionality of Elluminate in real time. The previous virtual learning environment was just text-based. The lecturer and students had to communicate using the written word without any sound or pictures and images to discuss and present: “Elluminate is light years ahead because the virtual learning environment (the previous software) was only text-based”.

There are more than just text-based interactions in Elluminate as graphics can be shared which is important to learning. In the graduate course, the students have only a 2-hour session each week and they have to do a lot of self study in their own time. The Elluminate session time is used to update, discuss, raise any issues that they do not understand or require additional clarification from the lecturer: “I use a lot of graphics to overview the course content to make sure the students have not got any queries and that they are happy with the week learning that they have to do.” This is crucial in a neurology course that looks at neurological processes through different scanning mechanism using CT scan and IMR: “...looking at different neurological processes through different scanning mechanisms is very crucial”. The discussion on the neurological processes was made clearer by showing actual CT scanned images and IMR graphics: “...able to do that in Elluminate by uploading CTS, IMR and those sort of things are very helpful”.

Elluminate also enables audio conferencing and up to six simultaneous speakers. In the neurology course, the lecturer allowed the maximum number of audio and video participants:

“I had maximum simultaneous talkers and maximum visual. I had all the pictures (video) of the students every week”.

However, if the students have audio problem, they could easily use the textbox feature that is also available in Elluminate to ask questions and provide feedback in discussion: “Some of the students have problems with sound and things so we use textbox”.

Elluminate also has several whiteboard interaction tools including a pointer and a highlighter for the virtual whiteboard. Items can be circled or coloured: “I use a pointer to actually point to different things as I go through because as I was going through a CT scan for example or a scanned picture, I use a pointer to point at the hotspot or the area that was significant”.

8.3 Technological Opportunities and Challenges

As with other technology implementation, there are pros and cons, the obstacles that had to go through before getting to launch it, the silver lining behind grey clouds. In Elluminate, the students that participate in a real time session drop out and in again: “people/computer dropping out”. In the neurology course case, the coordinator ran into a major problem in the initial stages. She was not able to login into the system. When she called Information Technology Support (ITS) staff, the support staff took a long time to solve the problem as he was not a participant of the pilot study and had no idea about Elluminate: “Initially I run into major problems in the beginning of the semester...the first call to ITS they had no more idea about Elluminate than I did (because they were not informed or included in the pilot study)”. The ITS staff and support staff from the teaching and learning unit were very helpful in trying to solve the problems and obstacles faced by the graduate course coordinator: “...(support) people have been very supportive”. She managed to get Elluminate running and has been using it ever since:

“The ability to talk is an advantage because virtual classroom (previous software), you could not talk, you could only text with typos and quick typing”.

With Elluminate, a complex-based web interactive multimedia technology, she could talk, interact with the students on the whiteboard: “I have them draw on the whiteboard” and “we do discussion verbally through microphone”.

She had already prepared the materials for the on-campus students that came to for a face-to-face lecture classes. Although the mode was different, she managed to use the same material for the online real time sessions: “I am prepared for the course for the on campus student. So now I am going (to use the materials) on Elluminate, They (the students) would have done the reading, it sort of just a bit different mechanism really. So I do not have to prepare anything other than what I would normally do”. Using the same resources, Elluminate provide advantages in adding interaction and functionality for building academic and student relationship with the students.

However, for the real time Elluminate session, the expectation is that the students have already done their weekly reading and learning and come to the 2-hours session

for further discussion and clarification for them to understand further and achieve the learning outcome intended for that week:

“The expectation is probably more self-directed learning because I got less time with them (web interactive multimedia technology students)”.

She managed to surmount the obstacles and use the web interactive multimedia technology (WIMT). Elluminate, was useful for the neurology graduate course coordinator and students that lived in different suburbs and were scattered across different states and countries. Elluminate also enabled real-time interaction with students. This was more than mere text exchange as the facilitator could communicate verbally, point to graphics and get polling and audio feedback from the students: “It is fantastic! I think it is a fabulous technology”.

The technology increased the opportunity for interactively align the objectives, activities and assessment. The number of students that she manages through the technology is small and it enables smooth working functions during the real time session on the web.

Example solutions taken by the university IT services to tackle the problems were creating user groups to faced the technological infrastructure problems and support each other challenges. However, some staff learnt to use the application on their own because it was relatively simple.

Organization thinking about adopting should proactively organize training and user group to enable practices among staff. Staffs need to use the application in a learning environment to assess the useful features and to decide how their curriculum, pedagogy and resources need to change.

These findings show that by implementing WIMT in the university, effective teaching and learning activities that use multimedia can occur across physical space and geographic boundaries. Elluminate in this case used several interactive multimedia features that enabled interaction that could be accomplished by more than just text exchange. The effective use of WIMT in learning, teaching and curriculum design requires a more informed understanding of the expectations of students, staff and institutions, along with preparation for and induction into the use of technology to foster positive learning and student outcomes [11]. The learning constructivism models learning as objective, activity and assessment aligned. Complec-based WIMT augment traditional online environment in the university by enabling more features that provided more opportunity for interaction in building the academic and student relationship.

9 Conclusion

In a pilot study, all stakeholders including front line technical support staff should be included and be introduced to the web interactive technology that was piloted and is currently being implemented in the university.

Access to and the ability to effectively use ICTs to obtain information and services are becoming increasingly important requirements necessary to fully participate in contemporary Australian economic, political and social life [15]. Eklund, et al., [15], stated that successful learning required quality instructional content as well as an appropriate context that includes facilitation and an understanding of the learner. The

sharing of images and applications enabled in WIMT provides this quality content and more interaction through pointers and highlighters. However, in this case, the learning is more self-directed. The learner is expected to explore first and then get further clarification and understanding from the real time session with the lecturer.

On using ICT, the educational theory has also had an impact as a theoretical basis upon which to justify content designs [15]. Biggs [21] principles of constructive alignment have fostered an academic environment where students can be confident that a course unit's learning outcomes line up with its learning objectives. In this case, the learning objectives and learning outcomes was provided to the students for their learning of the course.

The technology is seen increasingly as an enabler of learning. In the multimedia and web development industry there is a clear evidence of a gradual maturing of practices, through understanding of user centered design standards and the importance of usability in design [15]. These improvement processes are assisting to create better quality resources which are more efficiently produced and better meet the needs of the target market. The user in this case was able to easily adapt and use the technology without technical assistance or training.

The lecturer supervised the successful deployment and integration of the content into the teaching and learning environment. The lecturer's role was to find, adapt and deliver knowledge using a variety of techniques appropriate to a knowledge domain and the needs of the learner.

The evaluation of web-based learning environment was a continuing process throughout the development lifecycle [22]. Several evaluation approaches could be used to identify problem areas or to draw inferences about the overall quality of web-based learning environment. Several studies consider how effective the user interface system support users' learning activities. This research-in-progress will look at the use of the system features that support end-user's understanding and learning outcomes.

The dynamic nature of learning contexts and appreciation of the fact that even if the environment is stable each semester will be different due to the inherent diversity amongst student cohorts needs to be considered in evaluation processes [12]. Evaluation can be used as method for online education in university learning [23].

This paper provides an overview of Web Interactive Multimedia Technology and issues and opportunities for adoption in higher education. This system provides a flexible environment for academics and learners to communicate across physical space. In particular, the potential in a blended environment are emphasized. It enables learning beyond the traditional boundaries, and that the introduced system provides useful alternatives.

By looking at the WIMT implementation, further improvement to the design issues can be done by looking at the technical issues and integration of features to activities for effective learning. For a small number of students, Elluminate provides opportunities for geographically disparate groups and increases interactivity during learning actuates for online students.

The WIMT provides real time collaborative feedback between academic and student. It enables synchronous as well as asynchronous features (with its recording facility). This could enhance lifelong learning activity to additionally generate a knowledge based worker and society.

Acknowledgments. This research is part of a study supported by International Islamic University Malaysia and the Malaysia of Higher Education Bumiputra Academic Training Scheme.

References

1. Gomez, J.: Conceptual Modeling of Device-Independent Web Applications. *J. IEEE Multimedia* 8, 26–39 (2001)
2. Conte, T., Massollar, J., Mendes, E., Travassos, G.H.: Usability Evaluation Based on Web Design Perspectives. In: *First International Symposium Empirical Software Engineering and Measurement, ESEM*, pp. 146–155 (2007)
3. Cochrane, T.: Mobilising Learning: A Primer for Utilising Wireless Palm Devices to Facilitate a Collaborative Learning Environment. In: *ASCILITE Conference*, pp. 147–157 (2005)
4. Ivory, M.Y., Megraw, R.: Evolution of Web Site Design Patterns. *ACM Trans. Inf. Syst.* 23, 463–497 (2005)
5. Baharun, N., Porter, A.: Teaching Statistics Using a Blended Approach: Integrating Technology-based Resources. In: *ASCILITE Conference, Auckland, New Zealand*, pp. 40–48 (2009)
6. Boulay, R., Anderson, C., Parisky, A., Campbell, C.: Developing Online Training Materials in Molecular Biology: Enhancing Hands-on Lab Skills. In: *ASCILITE Conference, Auckland, New Zealand*, pp. 91–95 (2009)
7. Craig, P., Wozniak, H., Hyde, S., Burn, D.: Student Use of Web Based Lecture Technologies in Blended Learning: Do These Reflect Study Patterns? In: *ASCILITE Conference, Auckland, New Zealand*, pp. 158–167 (2009)
8. Cody, T.L.: Discovering Aesthetic Space Online? In: *ASCILITE Conference, Auckland, New Zealand*, pp. 153–157 (2009)
9. Iivari, J., Isomäki, H., Pekkola, S.: The User – The Great Unknown of Systems Development: Reasons, Forms, Challenges, Experiences and Intellectual Contributions of User Involvement. *Info. Sys. J.* 20, 109–117 (2010)
10. Arbaugh, J.B., Godfrey, M.R., Johnson, M., Pollack, B.L., Niendorf, B., Wresch, W.: Research in Online and Blended Learning in The Business Disciplines: Key Findings and Possible Future Directions. *The Internet and Higher Education* 12, 71–87 (2009)
11. Krause, K., McEwen, C.: Engaging and Retaining Students Online: A Case Study. In: *32nd HERDSA Annual Conference, Darwin*, pp. 251–262 (2009)
12. Gosper, M., Green, D., McNeil, M., Philips, R., Preston, G., Woo, K.: Impact of Web-Based Lecture Technologies on Current and Future Practices in Learning and Teaching. Report, Australian Learning and Teaching Council (2008)
13. Nelson, M.R.: Emerging Digital Content Delivery Technologies in Higher Education. Report, ECAR Research Bulletin (2006)
14. Heckman, R., Qing, L., Xue, X.: How Voluntary Online Learning Communities Emerge in Blended Courses. In: *39th Annual International Conference on System Sciences, Hawaii* (2006)
15. Eklund, J., Kay, M., Lynch, H.M.: E-learning: Emerging Issues and Key Trends: A Discussion Paper (2003)
16. Zhang, D., Zhao, J.L., Zhou, L., Nunamaker Jr., Jay, F.: Can E-Learning Replace Classroom Learning? *Commun. ACM* 47, 75–79 (2004)

17. Williams, J., Fardon, M.: Perpetual Connectivity: Lecture Recordings and Portable Media Players. In: *ICT: Providing Choices for Learners and Learning*, ASCILITE Conference, Singapore, pp. 1084–1092 (2007)
18. McElroy, J., Blount, Y.: You, Me and iLecture. In: *Who's Learning? Whose Technology?* ASCILITE Conference, Sydney, pp. 549—558 (2006)
19. Soong, S.K.A., Chan, L.K., Cheers, C.: Impact of Video Recorded Lectures Among Students. In: *Who's Learning? Whose Technology?* ASCILITE Conference, Sydney, pp. 789–793 (2006)
20. Yin, R.K.: *Case Study Research: Design and Methods*. Sage Publications Incorporated, Thousand Oaks (2009)
21. Biggs, J., Tang, C.: *Teaching for Quality Learning at University: What The Student Does*. Society for Research into Higher Education & Open University Press, New York (2007)
22. Nam, C.S., Smith-Jackson, T.L.: Web-based Learning Environment: A Theory-Based Design Process for Development and Evaluation. *J. of Info. Tech. Edu.* 6, 23–43 (2007)
23. Stigmar, M., Karlsudd, P.: On-line Education, More Than One-Way Education? *J. of Emerging Techn. in Web Intelligence* 1, 77–87 (2009)

Health Architecture Based on SOA and Mobile Agents

Mohamed Elammari and Tarek F. Alteap

Faculty of Information Technology, University of Garyounis, Benghazi, Libya
elammari@garyounis.edu, tarekfayez@garyounis.edu

Abstract. Mobile agent technology is used in modern information technology to facilitate interoperability between isolated information systems. In addition, mobile agents have many advantages that make them a viable and attractive option in health sector applications. We propose the development of a health architecture based on integrated mobile agent technology and a service-oriented architecture (SOA) with distributed health applications involving the interoperability of remote or local homogeneous and heterogeneous applications, besides the SOA, and providing support for mobile agents. Furthermore, the SOA can be tested on different platforms. The development of a mobile agent architecture using Health Level Seven, known as HL7, for data exchange in the health sector is an essential step in making it widely accepted by centers that use health records as their means of communication. Each health center's local data format can be modified for data exchange with other health centers through a mapping process that transforms the data from its local format to HL7, and vice versa. This research introduces a mobile agent architecture that can be applied to distributed health information systems to achieve interoperability.

1 Introduction

The current state of electronic health records (EHRs) is such that a mechanism is needed to regulate information exchange, so that different formats can be used for communication. The main problem with EHRs is that, until now, no architecture has described how such interoperability is to be achieved. This interoperability must be described within a standard environment so that it can facilitate the interconnectivity of different health organizations. Such an information exchange must naturally focus on EHR standards. This paper proposes to describe such an architecture, including not only its structure and behavior but also its usage, functionality, performance, resilience, reuse, comprehensibility, and technological approaches. It considers a general interoperability architecture, such as CORBA, and adjusts it for application to such specialized domains as health systems.

Such an approach has until now led to various problems, such as the lack of a common definition for health interoperability, duplication of effort, the inability to satisfy industrial-strength requirements, and incompatibility issues. Addressing these issues requires a health architecture that describes the interoperability between health centers with ease and clarity. Another problem that must be addressed is the lack of a collaborative architecture that uses mobile agents, a mistrusted application, within a

service-oriented architecture (SOA) framework. We are both encouraged and enthusiastic about our approach towards this goal.

In an effort to promote health interoperability, vendors have implemented certain EHR standards. Their implementation methods, however, have varied greatly, and this, especially combined with the limitations of presently available information standards [1], has resulted in the lack of system interoperability.

The World Health Organization has determined that to support patient care and the evaluation of healthcare services, the standard definitions of data exchange must address data integration. Among the accredited standards-developing organizations in the international healthcare industry, the most well known is Health Level Seven. Health Level Seven operates in the United States and developed HL7, the most widely used healthcare-related electronic data exchange standards in the world [2].

The primary framework of our architecture describes its implementation, and, for this purpose, an SOA is appropriate. In particular, one can implement an SOA to create a group of loosely coupled, reusable services that can be subsequently dynamically assembled and reassembled into different applications, according to ever-changing business requirements [3]. If such an SOA is properly implemented, it could harmoniously incorporate the nature of various mobile agents, and allow faster development and deployment of enterprise applications, for less cost than previously possible.

This paper develops a health architecture that is based on integrated mobile agent technology and an SOA with distributed health applications to facilitate the interoperability of remote and local applications, both homogeneous and heterogeneous. The SOA should be capable of supporting mobile agents and should be tested on different platforms, such as .NET, depending on the SOA. Building and deploying a system using a mobile agent or mobile agent system and SOA technology is too complex a task for any systems programmer. Therefore, for resiliency, this complexity must be addressed by the architecture itself; that is, the architecture's components and construction must work together to exchange health information.

This paper proposes a mobile agent health architecture that defines and describes a standard environment for health record interoperability among health organizations. Its main goal is for the proposed mobile agent health architecture to act as a guide for developers and programmers to achieve practical applications within its context.

2 Background

A system infrastructure's architecture describes its components and their high-level interactions with each other. The components are abstract modules that are built as a "unit" with other components, and their high-level interactions are connectors. The configuration of these components and connectors describes the system's structure and behavior. The Institute of Electrical and Electronics Engineers also defines an architecture as "the highest level concept of a system in its environment." By offering generic, reliable services as application building blocks, an SOA utilizes various methods and technologies to dynamically connect software applications between

different platforms and business partners, thus allowing enterprises to communicate with each other. For example, Web services are one of the important SOA technologies [3]. Because mobile agents are software processes that are not constrained by either client–server or subprogramming communication mechanisms, they are free to move to the actual destination of the required service. They can do so even after their first migration, since the code does not have to be split, where one part resides on the server and the other on the client. All the code exists in a single object that can move about as needed. There is therefore no client–server relationship, since the objects themselves move, and all computers are peers. A mobile agent can have a home or return data to its caller, but this situation is completely dynamic, depending only on the choice of agent and not on the overall design [9]. Some of the advantages of mobile agents are a reduction in network load and latency, protocol encapsulation, asynchronous and autonomous execution, and fault tolerance.

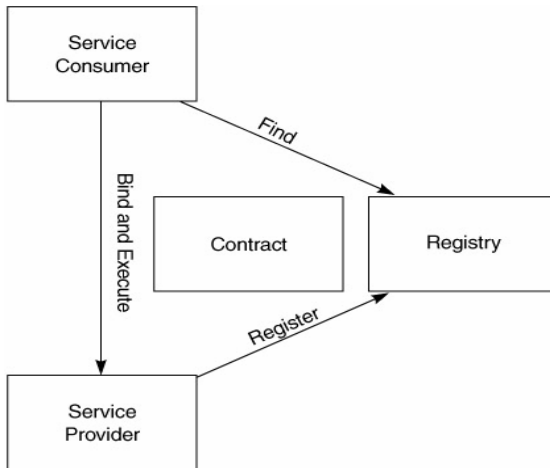


Fig. 1. SOA entities [4]

The various SOA entities [4]—consumer, provider, contracts, and registry—maximize loose coupling and reuse, shown in Fig. 1. Although described here in an abstract fashion, the following service entities are implemented in the SOA. First, the service consumer is represented within the structure of the entire World Wide Web and by the way the service executes. A local health service (LHS), for example, is a task provided by the service consumer. Second, the service provider is a network-addressable entity that accepts and executes all consumer requests. Third, the service registry is an entity that accepts and stores service providers’ contracts and provides them to the service consumers as needed. Finally, a service contract defines a set of pre- and/or post-conditions and those services that a service consumer should receive from the service provider.

3 Mobile Agent Health Architecture

Our proposed architecture aims to provide a method of communication among distributed health systems that wish to share information. It depends on the idea of

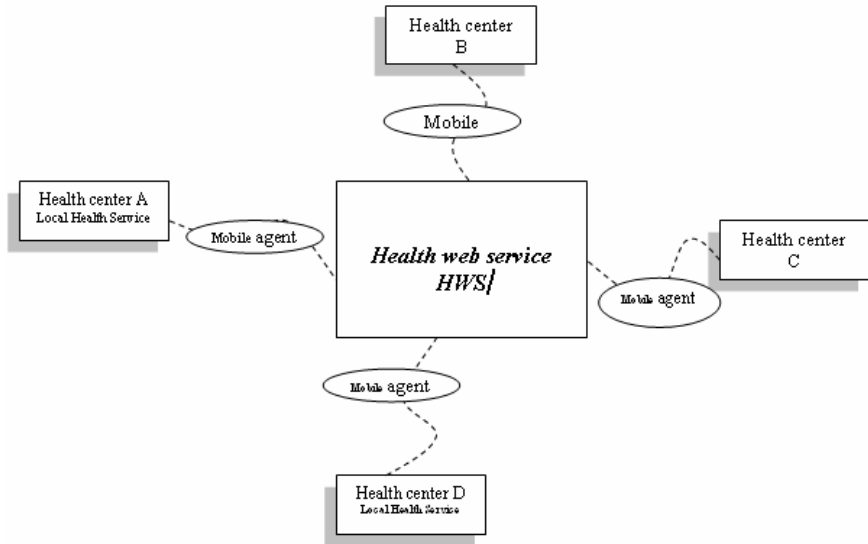


Fig. 2. Conceptual models for the mobile health service

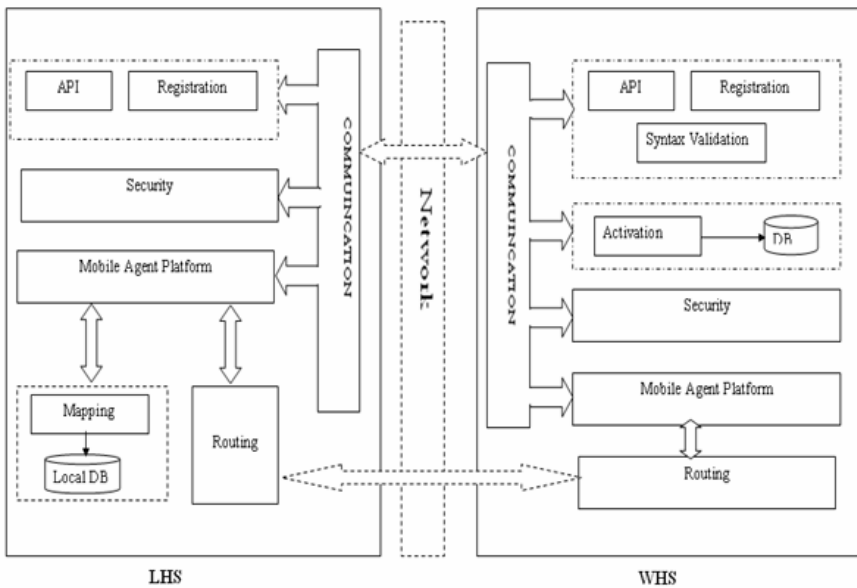


Fig. 3. Health mobile agent architecture

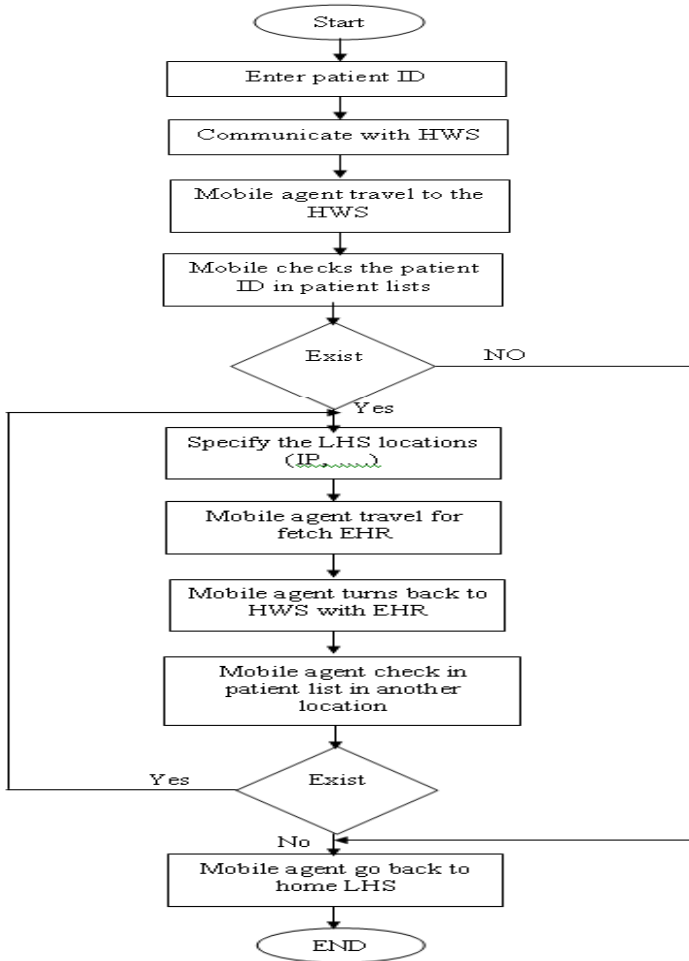


Fig. 4. The steps to obtain EHR

providing a service that allows such systems to exchange information amongst themselves, this service being a health web service (HWS), and that achieves interoperability with each system's local health service (LHS). A mobile agent is triggered that goes between the HWS and the LHS that resides on a health center's host, with the HWS directing the mobile agent's routing so that it can obtain the needed information. The basic idea of this architecture is to link various health centers through the HWS, playing a key role in the communication process, so that health centers do not have to be in direct contact with each other and every site has an LHS specified to deal with health information garnered through the HWS (see Fig. 2).

The HWS supports a mobile agent that travels between different medical sites participating in the service to perform its tasks. Within this context, certain services must be available to both the HWS and the medical site to help the mobile agent successfully gain information. To establish the architecture, careful consideration needs to be given to the requirements of such a system.

For further clarification, we illustrate the main activities of the architectural process in a step-by-step flowchart. We can then examine each individual step more fully, without being overwhelmed by the bigger picture, leading to a better understanding of the architecture. Fig. 3 illustrates the architecture as a whole.

The flowchart in Fig. 4 can be used to define activities that are explicitly required in the architecture.

4 Architecture Components

4.1 Authority Application Programming Interface

The authority application programming interface (API) refers to the application interface that enables an authority to interact with and manage its mobile agents, both locally and remotely, including such tasks as creating agents, communicating with them, and destroying them[5] (Fig. 5).

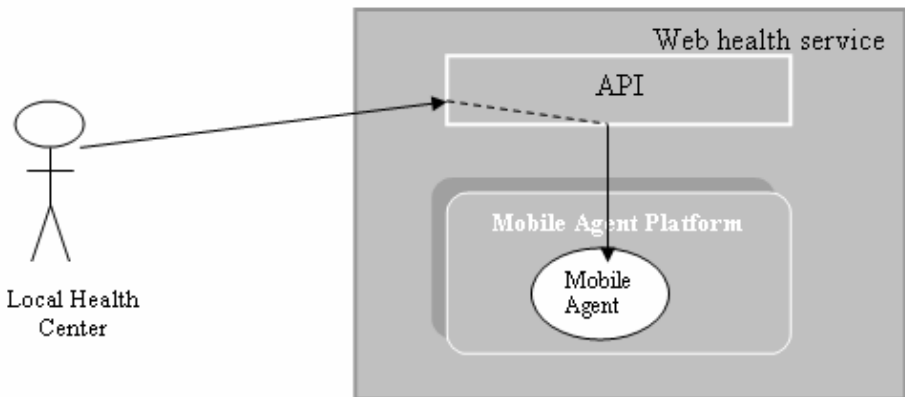


Fig. 5. Local Health Center interact with mobile agent

4.2 Mobile Agent Platform

The mobile agent platform refers to the environment on a host or a site where mobile agents can be created, terminated, or suspended through an authority API. [7]. The agent platform, shown in Fig. 5, is responsible for administering mobile agents. Mobile agents can be further classified as manager agents that perform management tasks; agents that facilitate applications, facilitate communication, monitor mobile agents, or log events; repository agents that retrieve and add information and query repositories; and interface agents, which provide the necessary API and interface with other entities and applications. Participants' agents provide a support system for executing cooperative processes [5].

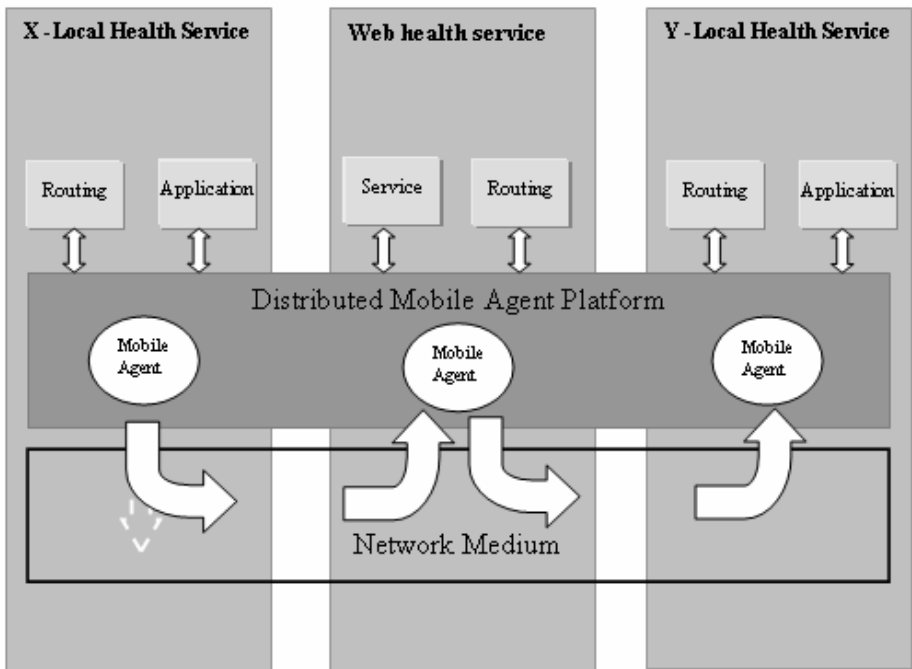


Fig. 6. Mobile agent platform

4.3 Registration and Activation

The first step, an LHS must register to become a member of a health service. It sends the required data to the HWS to be saved, authenticated, and given an LHS unique ID. An acknowledgment is then sent back that confirms registration and also indicates the time limit the LHS has to send patients' IDs to be registered.

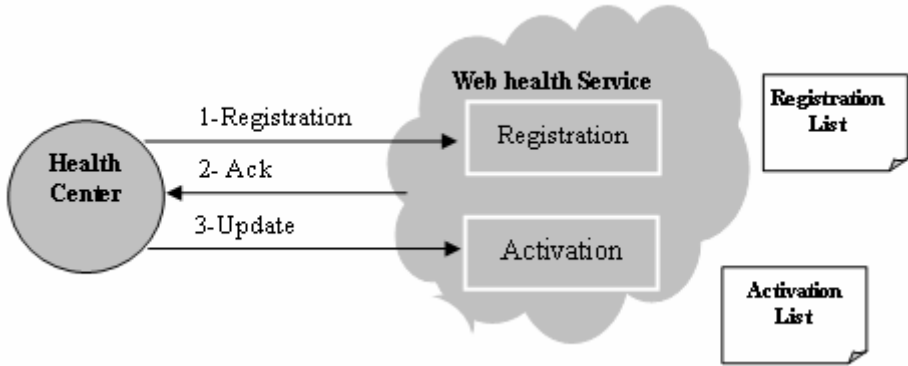


Fig. 7. Registration and Activation of Local Health center

After the LHS register the patient IDs in its database, it must periodically perform updates to keep these registrations active. The steps are shown in Fig. 7.

4.4 Routing Services

The HWS has network information about each health center registered in the service, so that the health center ID can provide the routing service with the health center's network information, such as its IP address and port number.

4.5 Communication

One very important issue is that a communication mechanism is required to collect all the types of messages passed between the LHS and the HWS. Fig. 8 shows the types of communication messages. One of the characteristics of the HL7 standard used here is its ability to create composite message structures to indicate health care needs. These messages can be coded in AL1, OBX, DG1, and ORC and indicate laboratory test results, imaging studies, nursing and physician observations, and pharmacy and drug use.

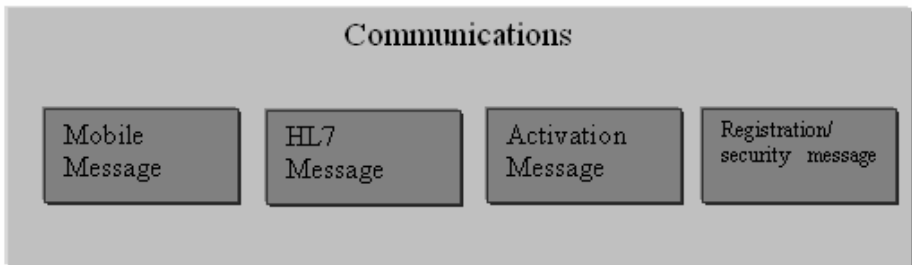


Fig. 8. Type of communication messages

4.6 Security

Security is a major factor in the success of any network application and we therefore consider the following techniques for securing LHS and HWS data (Fig. 9).

1. Single-source data and the tools to use the data allow a local health center to accept only the mobile agent from a single site (HWS), as specified by an IP address and the DNS.
2. A public key infrastructure (PKI) is used for the safe interoperability between the LHS and HWS as follows:
 - 2.1. Upon its first connection with the HWS, the LHS generates a public/private key combination.
 - 2.2. The LHS share its public key with the HWS.
 - 2.3. The LHS encrypts its identity information, such as its name, with its private key.
 - 2.4. The LHS sends both the identity information and the encrypted identity information to the HWS.
 - 2.5. The HWS decrypts the encrypted identity information and compares it with the clear-text identity information sent by the LHS.

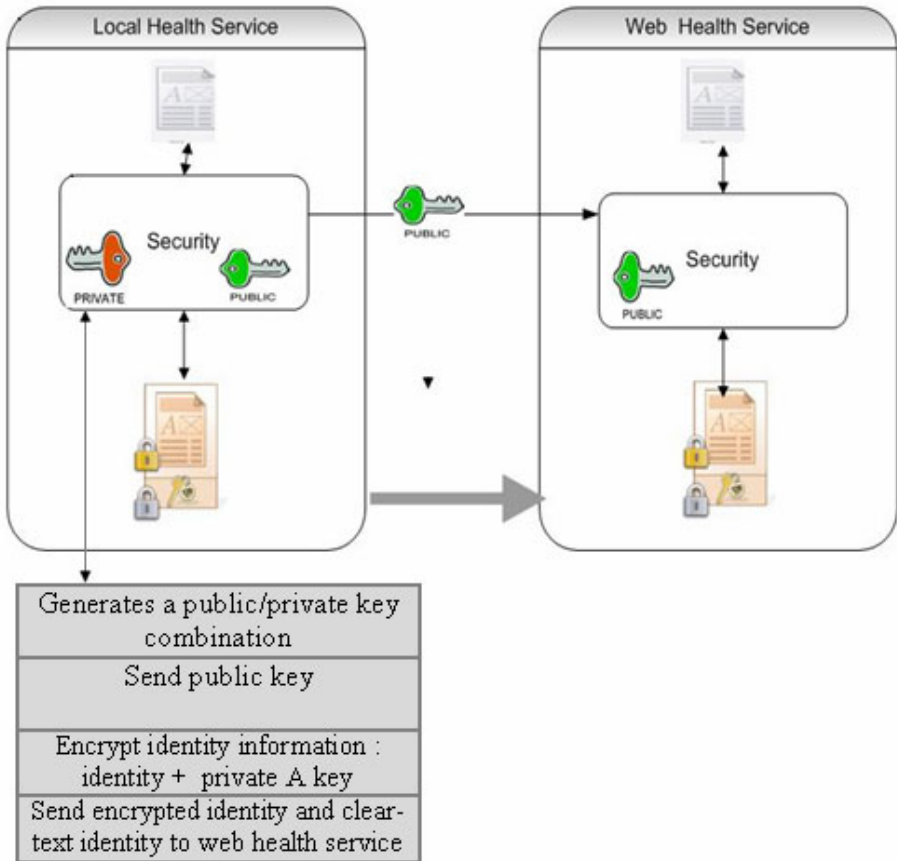


Fig. 9. Security using PKI for safe communications

4.7 Mapping

The goal of this architecture is to exchange health information between the LHSs throughout an HWS and therefore requires an intermediary data format that is accepted and understood by all the health centers within the system. We choose the standard HL7 message format as the common language between health centers, since these centers are able to transform any health information from their local formats to the HL7 format, and vice versa. The service provides the mapping for this transformation. Mapping from the source to the target is accomplished with the help of mapping tools that produce a mapping definition. These tools carry out what is called normalization mapping, which describes how a specific message is transformed into the corresponding schema. During the installation of the mapping tool in an LHS, a matching process between the source and HL7 messages takes place wherein the LHS defines the path for the local database and links the database's fields with the corresponding fields in HL7, thereby allowing a transformation of the database's source fields into HL7 messages, and vice versa (Fig. 10).

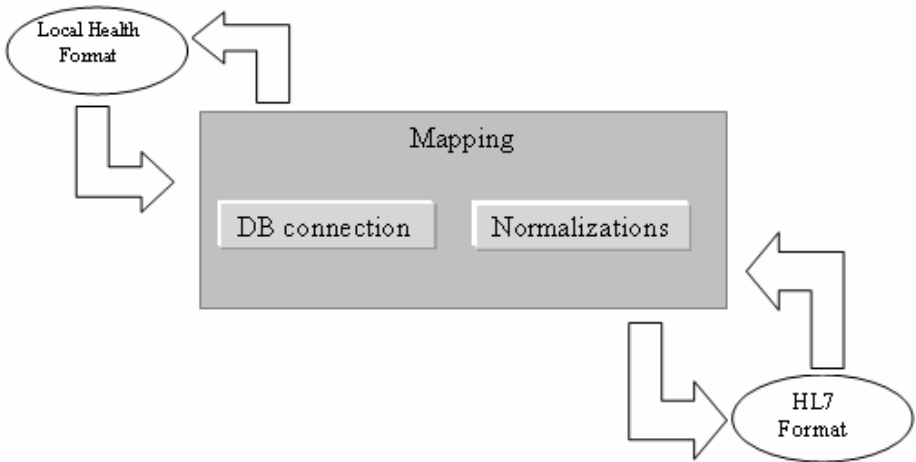


Fig. 10. Mapping Transformation process

5 Evaluating the Mobile Agent Health Architecture

Table 1 summarizes the most important measures of the proposed architecture based on evaluation criteria proposed in [4].

Table 1. Characteristics of the proposed mobile agent health architecture

Characteristics	Answer (Yes/No)	Comment
Does the service provide rules or decisions?	Yes	Periodic updates
Does the service have a specification?	Yes	Post-conditions
Does the service conform to enterprise standards for the type of service?	Yes	HL7, SOA
Does the service provide a single, consistent way to access data or perform a business function?	Yes	Contact via the Web only
Is the use of this service appropriate to the purpose?	Yes	Communication via the Internet and Web service technology unknown and spread for the all
Is the scope and granularity of the service appropriate for its purpose?	Yes	The mobile agent suitable to work with the Web, can handle network disconnections
Does this service adhere to the security requirements of the enterprise	Yes	PKI
Is each service operation responsible for a discrete task?	Yes	In the design, we tried to adhere to standards
Does this service limit what the consumers must know to invoke it in the way described in the service specification?	Yes	The consumer must only know the requirements to run the LHS
Does the service avoid making assumptions about the purpose or business characteristics of the consumer?	No	The service makes some assumptions
Does the service use other services and components to achieve functionality?	No	
Is the service's granularity appropriate for its intended use?	Yes	Mobile agent are effective, improving application latency and bandwidth and reducing vulnerability to network disconnection

6 Conclusions and Future Work

The development of this health architecture is based on integrated mobile agent technology and an SOA with distributed health applications to achieve the interoperability of remote or local homogeneous and heterogeneous applications, besides the SOA, with support for the mobile agents. Furthermore, the SOA can be tested on different platforms. We find this to be a good approach to building an architecture that introduces the services that an organization provides to its clients, customers, communicating partners, and other organizations, services that are fundamental to the needs of the business. The SOA brings competitive advantages to enterprises in the sense that these services can easily react to changing business requirements and allow one to structure an agile and responsive system. This

architecture also provides a mechanism and standard environment for interoperability services between health organizations based on integrated mobile agent technology and an SOA with a distributed health application. The elements of such a mobile agent health architecture provide a flexible standard environment involving all the services needed to communicate between health organizations. The mobile agent platform plays a key role in controlling and manipulating the mobile agent and the HWS, facilitating connectivity to each LHS. This mobile agent health architecture provides a common platform and execution environment for heterogeneous health applications built with diverse technologies. Service orientation supports the interoperability of these applications by hiding their internal structures from each other.

The architecture in each stage and the design of individual services can be further developed to make the implementation and creation of applications easier for the developer. Further study should also examine grid computing to exploit its benefits within the architecture.

References

1. Harris, M.R., Ruggieri, A.P., Chute, C.G.: From Clinical Records to Regulatory Reporting: Formal Terminologies as Foundation. *Health Care Financing Review* 24(3), 118 (2003)
2. Electronic Health Records Overview, National Institutes of Health National Center for Research Resources. MITRE (April 2006)
3. McGovern, J., Tyagi, S., Stevens, M.E., Mathew, S.: *Java Web Services Architecture*. Morgan Kaufmann, San Francisco (2003)
4. Rosen, M., Lublinsky, B., Smith, K.T., Balcer, M.J.: *Service-Oriented Architecture and Design Strategies*. Wiley, Chichester (2008)
5. Schoeman, M., Cloete, E.: Architectural Components for the Efficient Design of Mobile Agent Systems. In: Proceedings of the 2003 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology (2003)
6. Da Silva, A.R., Da Silva, M.M., Romao, A.: Web-Based Agent Applications: User Interfaces and Mobile Agents. In: Proceedings of the IS&N 2000 Conference (2000)
7. Wang, A.I., Hanssen, A.A., Nymoen, B.S.: Design Principles for a Mobile, Multi-Agent Architecture for Cooperative Software Engineering. *Software Engineering and Applications* (2000)
8. Shiva, S.G.: *Computer Design and Architecture*, 3rd edn. CRC Press, Boca Raton (2003)
9. Hower, C.Z.: *Mobile Agents -Software on the Move*. The Code Project (2005)

Metrics Based Variability Assessment of Code Assets

Fazal-e-Amin, Ahmad Kamil Mahmood, and Alan Oxley

Computer and Information Sciences Department, Universiti Teknologi PETRONAS,
Bandar Seri Iskandar, 31750 Tronoh, Perak, Malaysia
fazal.e.amin@gmail.com, {kamilmh, alanoxley}@petronas.com.my

Abstract. The potential benefits of software reuse motivate the use of component based software development and software product lines. In these software development methodologies software assets are being reused. Variability management is a tenet of software reuse. Variability is the capacity of software to satisfy variant requirements. Variability, being the central player in reuse and an important characteristic of reusable components, needs to be measured. In this paper we acknowledge this need and identify measures of variability. Variability implementation mechanisms are analyzed followed by metrics. The metrics are applied on open source component code and the results are validated by an experiment carried out with human subjects.

Keywords: Software variability, metrics, variability implantation.

1 Introduction

Software reuse is the process of using existing artefacts to develop new software. ‘Reusability’ is a software quality factor. Software reusability is the “degree to which a software module or other work product can be used in more than one computer program or software system” [1]. It is also defined as the property of software that relates to the probability of reusing it [2].

Software reuse is one of the factors that helps to reduce the time and effort required to develop and maintain software systems [3]. Research on the effects of reuse on the development process have been briefly summarized in [4]. Some of the research described looks at the relationship between reuse, quality and productivity and the results showed that software reuse results in better quality. In another study [5], it is stated, on the basis of industrial data, that software reuse significantly increases productivity and reduces the time spent on correction efforts. In [6] research is conducted on reuse at NASA, and it mentions that reuse is common in development teams contracted by NASA. The prime motivations for reuse by these teams are the saving of time and the reduction of cost. The authors identify different types of reuse: accidental, adaptive, black-box, horizontal, systematic, vertical and white-box.

Software reuse can be commonly employed in two ways. One way is the use of component libraries where the components are available to reuse. The second way is systematic reuse in the form of software product lines. Component libraries contain generic and small components; on the other hand, product lines deal with domain specific components.

Variability is defined as the “degree to which something exists in multiple variants, each having the appropriate capabilities” [7]. Variability of a software component is related to its reusability. An increase in variability increases the likelihood of its reuse.

2 Software Variability

Variability management (VM) is an important activity in a reuse intense software development environment. It is a non trivial activity and has many facets as not only are both architecture and coding variable, but so also is the development process, as different tools can be used.

VM in a product line context refers to the identification, modeling, resolution, storage and instantiation of variability [8]. VM is the distinguishing feature of software product line development [9]. Efficient VM is one of the key success factors in a reuse intense software development environment. In product line development, all the artifacts developed are considered core assets. Variability is considered as a characteristic of a reusable core asset [10].

In [11] variability types are defined. The types include attribute, logic, workflow, persistency, interface, and combined. As regards attribute variability, an attribute is supposed to be a placeholder for values to be stored – such as constants, variables or data structures. Three cases are presented. First is when the number of attributes varies between members of a product family. Second is the variation in the data type of the values assigned to the attributes, and the third case represents the variation of the value assigned to the attribute that is persistent.

Logic variability is the variation of the algorithm or logical procedure. There are several cases of logic variability, each case dependent upon the entity that varies, be it the procedural flow, the post condition, the exception handling, or the side effects between family members. Workflow variability is variation in the order, type and number of methods invoked by family members when carrying out a common task.

Persistency variability refers to the variation in the values of attributes that are stored in secondary storage. Interface variability is the variation in the signature of the interface method, i.e. to implement the same requirement, different members of a family implement their methods in different ways. These are distinguished by the name, return type, and order and type of parameters. Combined variability is where a variation point has more than one variability type.

In [12] variability is categorized as follows: positive - when some functionality is added; negative - when there is a withdrawal of functionality; optional - when code is added; alternative - when code is removed; function - when functionality is changed; platform/environment - when the platform or environment is changed.

Our focus during this discussion is on mainstream product line implementation technology, such as object oriented development, and specifically implementations based on the Java language. Another point to consider at this stage is that VM of requirements, design artefacts, and test cases is out of the scope of this study.

The term 'variability realization technique' refers to the mechanism which is used to implement at the variation point [13]. A variation point specifically identifies the part of a variable requirement that is subject to change. A variant is an instance of a variable requirement. A variant can be implemented in different ways, affecting different software entities and these entities may include components, classes, a set of classes or lines of code[13]. Variability can be introduced at different stages of software development such as during architectural design, during detailed design, during implementation, and when compiling or linking [13]. Different software entities are relevant at each of these levels. Since our work is concerned with implementation level variability, the software entities which we will focus on are individual classes and lines of code.

In the following sections, variability realization techniques are discussed based on [14] and [15].

A systematic review [16] presents the state of the art in the area of software measurement. The results of the review show that there is no measure available for variation. This shortage of metrics to measure variability, specifically at the implementation level, is also recognized in another study [17]. In our work we acknowledge this gap and propose metrics to assess the variability of software components.

3 Measurements in Software Engineering

In software measurement, three kinds of entities are measurable - processes, products, and resources [18]. A product can be defined as any artifact developed as a result of process activity. These entities may have attributes which are of two kinds - internal and external. An external attribute is one that cannot be measured directly. In contrast, internal attributes can be measured directly. If we can measure something directly then this means that we can measure it independently. Relevant metrics are termed 'direct metrics' [19]. For example, the size of a program can be measured directly in several ways: by counting the number of lines of code; by counting the number of 'methods'; etc. In software engineering measurement terminology, a metric is a quantitative indicator of a software attribute; a metrics model specifies relationships between metrics and the attributes being measured by these metrics. Another dimension in this field is the definition of metric as being elementary, in that it requires only one attribute, or composite, in that it needs more than one attribute[20]. The metrics defined in this paper fall under the category of product metrics as the work described measures code.

4 Research Methodology

The methodology comprises of the following steps:

- 1- Variability implementation mechanisms were identified.
- 2- Mechanisms were mapped to the types of variability.
- 3- Mechanisms were mapped to the features and scope of variability.

- 4- Well known and established object oriented metrics were used to assess variability.
- 5- Human assessment of variability was conducted by using a questionnaire.
- 6- Results of steps 4 and 5 were compared in an effort to establish the suitability of individual metrics for the measurement of variability.

Steps 1 to 3 involved a careful analysis of the variability implementation mechanisms. Details of these works can be found in [21] and [22]. In step 4 metrics were proposed. A brief description of the metrics used is provided in next section. These metrics are obviously selected due to the fact that they measure attributes that relate to variability.

5 Proposed Metrics

In [11] types of variability are defined on the basis of component reference models, namely CORBA and EJB. The building blocks of a component are defined as classes, work flow among classes, and interfaces.

We can consider the entities involved in object oriented programming. In Java these comprise the classes, interfaces, packages and Java beans. From the viewpoint of reuse, using Java beans is considered to be a black box approach. However, our work is concerned with a white box approach to the reuse of components.

An object oriented class consists of attributes, which hold data, and methods that exhibit behavior. An abstract class is used as super-class for a class hierarchy; it cannot be instantiated.

In [11] the following variability types are listed: attribute, logic and workflow. Another view of variability types is presented in [12] where variability is categorized as positive, negative, optional, function and platform/environment. All of the variability types given in [11] can be mapped to the variability types given in [12], for instance, the 'attribute' variability type is a 'positive' variable type when a new attribute is added.

Attribute variability can be implemented using any of the following techniques: inheritance; aggregation; parameterization /generics; overloading. Further cases of attribute variability are defined in [11]. One of these is the variation in the number of attributes. This type of variability is supported by inheritance and aggregation. Another type of attribute variability is variation in the data types of the attributes; this variability is supported by parameterization/generics.

As described earlier, inheritance is one of the mechanisms to handle attribute variability. In our work we propose variability metrics on the basis of the theory and mechanism of inheritance.

With inheritance the subclass inherits all the methods and attributes of the super-class. The subclass can define its own attributes in addition those it inherits from the super-class, which causes the attribute variability. The other mechanism associated with inheritance is overloading which causes logic and work flow variability. So, a class that is higher in the hierarchy, and therefore having more accessible attributes and methods, has more variability.

Goal: Assessment of object oriented systems to predict variability from the view point of the developer.

1. How much variability is there in the component?
 - 1.1. What is the ratio of method per class?
 - 1.1.1. Number of methods ÷ Total number of methods in component
 - 1.2. What is the ratio of number of child per class?
 - 1.2.1. Number of child ÷ Total number of classes

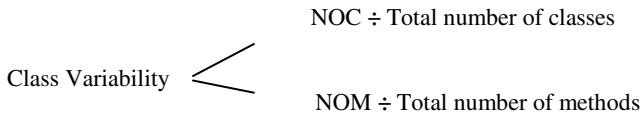


Fig. 1. Relationship of variability with metrics

The Number Of Children (NOC) metric is defined in [23] and the Number Of local Method metric is defined in [24]. The proposed metrics make use of these established metrics and associate the concept of variability with them.

6 Validation of Proposed Metrics

As with other engineering disciplines, software engineering is intended to help humans in solving their problems [25]. Software engineering, being a multidisciplinary field of research, involves issues raised by technology and society (humans). Software engineering activities depend on tools and processes. However, due to the involvement of humans, social and cognitive processes should also be considered [26]. Validation of new tools and processes is a necessary part of the advancement of software engineering [27]. The involvement of humans in software engineering demands the usage of research methodologies from the social sciences. Therefore, to validate the set of metrics selected to measure variability, a survey was used. A survey can be defined as a comprehensive system for collecting data using a standardized questionnaire [28, 29]. The information collected from a survey is used to “describe, compare or explain knowledge, attitudes and behavior” [28]. This type of validation is used in [30], where the term ‘experiment’ is used for the process of assessment of software (classes) by experienced developers and students. In [31] a ‘rating committee’ is used. A questionnaire is used in [32] and [33] for the purpose of validation of results.

Survey research is common in the software engineering discipline. Due to the effectiveness of surveys in software engineering, researchers have laid down a process to conduct surveys. A ten step process to conduct a survey is presented in [28]. In [29] a comprehensive seven step process for conducting a survey is explained.

We have used the approach presented in [29] and customized two steps. The details and the rationale for our decision are stated later in this section. The specific steps taken to conduct this variability assessment survey were:

- Identification of aim
- Identification of target audience
- Design of sampling plan
- Questionnaire formulation
- Pilot test of questionnaire
- Questionnaire distribution
- Analysis of the results

Let us clarify the purpose of this exercise. Our notion of a survey resembles the process used in [30] where, as we stated above, the term ‘experiment’ is used to conduct the assessment of software code by humans. In this paper we have used the term ‘survey’ because we are using the questionnaire as a tool to assess the code.

The aim of this survey is to get an objective assessment, from humans, of selected software code. Turning to the second step, 54 students of a software engineering class were asked to assess the variability of classes in the class hierarchies. Out of 54, five samples were discarded due to lack of information. The selected students had knowledge and experience in Java programming, software engineering, and the concept of object-orientation. They were studying these subjects as part of a degree program. A total of 15 classes were selected in three hierarchies related to three different components. Three components were selected, namely Component A from a rental domain, Component B from a computer user account domain and Component C from a bank account domain. More details of the components are provided in table-1. The components selected for this purpose were from Merobase (<http://www.merobase.com>). Merobase is database of source code files. The collection has more than 10 million indexed files, out of which eight million are Java files. A search and tagging engine is included.

Table 1. Component specifications

Component	No. of classes	No. of methods	Lines of code
A	03	31	204
B	03	11	78
C	09	34	281

A sampling plan was designed to decide the kind of statistical test used to interpret the results. The questionnaire was formulated and reviewed by the authors. The questionnaire was pilot tested and revised. The survey was conducted in two sessions, 18 respondents completed the questionnaire in the first session and 36 in the second session. Both sessions were conducted in the presence of the authors. The results of the survey were analyzed using statistical software.

7 Results

The response of the users was collected using a Likert scale from 1 to 5 - strongly disagree (1); disagree (2), neither agree nor disagree (3); agree (4); strongly agree (5).

The evaluators were asked eight questions to assess the variability of components. The arithmetic means of the responses is presented in table 2. The variability mean is the mean value of the means of individual responses to the questions.

Table 2. Results of variability assessment by human evaluators

Component	Variability (mean)
A	1.91
B	1.82
C	1.72

Calculation of variability

Class(x) Variability = (NOM of Class(x) / Total number of methods) + (NOC Class(x) / Total number of classes)

Calculation of variability for component B

Variability Class 1= (2/11) + (0/3) = 0.18

Variability Class 2= (4/11) + (0/3) = 0.36

Variability Class 3= (5/11) + (0/3) = 0.45

Avg. of Variability = (0.18 + 0.36 + 0.45)/3 = 0.33 (adjusted to compare with user assessment values

0.33 * 5= 1.65)

The variability values obtained by using the metrics are adjusted by comparing with the evaluator’s values. As described earlier, the values are with respect to the Likert scale from 1 to 5. The results are compared in figure 2. The results show that the metrics have successfully assessed the variability of component A and B. There is a difference in the values of component C. Further investigation is required to know the reasons for this difference.

Table 3. Results of variability assessment by proposed metrics

Component	Variability assessed by proposed metrics
A	1.66
B	1.65
C	1.14

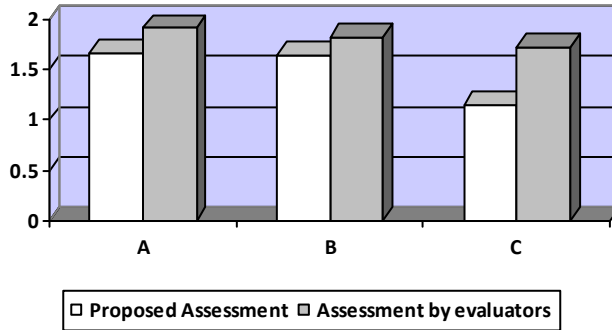


Fig. 2. Comparison of results

8 Discussions

The results in fig. 2 show both the variability assessed using the proposed metrics as well as that given by evaluators. Although only three components were studied, the amount of effort to do this is considerable. This is because of the effort required by each of the numerous evaluators to manually study the programming code in fine detail. The comparison in fig. 2 shows that the values for component A and component B are relatively close, which provides a reason for further investigations. In the case of component C, there is a significant difference. One of the reasons for this difference is the size of the component. Component C is the largest component, comprising of 9 classes, 34 methods and 281 lines of code. The evaluators perceived that this component has greater potential for variability because it has more classes. However, the values of variability obtained by the proposed metrics show that this component has less potential for variability. An increased number of classes do not mean increased variability. Furthermore, an increased size of a component does not guarantee increased variability. The proposed metrics were devised as part of our study and analysis of variability implementation mechanisms [22]. The proposed metrics only cater for the openness of a component for the variability mechanisms, regardless of its size, i.e. number of classes.

9 Conclusions

This paper concerns the measurement of one aspect of software quality. The concept of variability has an important place in reuse intense software development environments yet there is a lack of measures to assess it. This paper tries to fill this gap by introducing a new metric. An initial application of metrics and their validation shows successful results and suggests that more research in this direction is warranted. However, the new metric has only undergone a limited evaluation and more investigation needs to be done for a number of components of differing sizes.

References

1. IEEE: IEEE Standard Glossary of Software Engineering Terminology, NY, USA (1990)
2. Frakes, W.B., Kyo, K.: Software reuse research: status and future. *IEEE Transactions on Software Engineering* 31, 529–536 (2005)
3. Krueger, C.W.: Software reuse. *ACM Comput. Surv.* 24, 131–183 (1992)
4. Frakes, W.B., Succi, G.: An industrial study of reuse, quality, and productivity. *J. Syst. Softw.* 57, 99–106 (2001)
5. Mohagheghi, P., Conradi, R.: Quality, productivity and economic benefits of software reuse: a review of industrial studies. *Empirical Softw. Engg.* 12, 471–516 (2007)
6. Orrego, A., Mundy, G.: A study of software reuse in NASA legacy systems. *Innovations in Systems and Software Engineering* 3, 167–180 (2007)
7. Firesmith, D.: Common Concepts Underlying Safety, Security, and Survivability Engineering, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA (2003)
8. Schmid, K., John, I.: A customizable approach to full lifecycle variability management. *Science of Computer Programming* 53, 259–284 (2004)
9. van der Linden, F., Bosch, J., Florijn, G., Greefhorst, D., Kuusela, J., Obbink, J., and Pohl, K.: Variability Issues in Software Product Lines, in *Software Product-Family Engineering*, Vol. 2290, Springer Berlin / Heidelberg, pp. 303–38 (2002).
10. Her, J.S., Kim, J.H., Oh, S.H., Rhew, S.Y., Kim, S.D.: A framework for evaluating reusability of core asset in product line engineering. *Information and Software Technology* 49, 740–760 (2007)
11. Kim, S.D., Her, J.S., Chang, S.H.: A theoretical foundation of variability in component-based development. *Information and Software Technology* 47, 663–673 (2005)
12. Sharp, D.C.: Containing and facilitating change via object oriented tailoring techniques. In: *First Software Product Line Conference*, Denver, Colorado (2000)
13. Svahnberg, M., Gorp, J.v., Bosch, J.: A taxonomy of variability realization techniques: Research Articles. *Softw. Pract. Exper.* 35, 705–754 (2005)
14. Gacek, C., Anastasopoulos, M.: Implementing product line variabilities. *SIGSOFT Softw. Eng. Notes* 26, 109–117 (2001)
15. Pohl, C., Rummler, A., Gasiunas, V., Loughran, N., Arboleda, H., Fernandes, F.d.A., Noyé, J., Núñez, A., Passama, R., Royer, J.-C., Südholt, M.: Survey of existing implementation techniques with respect to their support for the practices currently in use at industrial partners. In: *AMPLE Project deliverableD3.1* (2007)
16. Gómez, O., Filipe, J., Shishkov, B., Helfert, M., Oktaba, H., Piattini, M., and García, F.: A Systematic Review Measurement in Software Engineering: State-of-the-Art in Measures, In: *Software and Data Technologies*, Vol. 10, Springer Berlin Heidelberg, pp. 165–76 (2008).
17. Mujtaba, S., Petersen, K., Feldt, R., Mattsson, M.: Software Product Line Variability: A Systematic Mapping Study. In: *15th Asia-Pacific Software Engineering Conference (APSEC 2008)* (2008)
18. Fenton, N., Pfleeger, S.: *Software Metrics: A Rigorous and Practical Approach*, PWS Publishing Co (1997)
19. IEEE: IEEE Standard for a Software Quality Metrics Methodology (1998)
20. Abreu, B.F., Goulao, M., Esteves, R.: Toward the design quality evaluation of object-oriented software systems. In: *Proceedings of the Fifth International Conference on Software Quality*, pp. 44–57 (1995)

21. Fazal, E.-A., Mahmood, A.K., Oxley, A.: Mechanisms for managing variability when implementing object oriented components. In: National Information Technology Symposium (NITS), King Saud University, KSA (2011)
22. Fazal, E.-A., Mahmood, A.K., Oxley, A.: An analysis of object oriented variability implementation mechanisms. SIGSOFT Softw. Eng. Notes 36, 1–4 (2011)
23. Chidamber, S.R., Kemerer, C.F.: A Metrics Suite for Object Oriented Design. IEEE Trans. Softw. Eng. 20, 476–493 (1994)
24. Li, W., Henry, S.: Maintenance metrics for the object oriented paradigm. In: Proceedings of First International Symposium on Software Metrics, pp. 52–60 (1993)
25. Jackson, M.: The Name and Nature of Software Engineering. In: Advances in Software Engineering: Lipari Summer School 2007, Lipari Island, Italy, July 8-21, 2007. Revised Tutorial Lectures, pp. 1–38. Springer, Heidelberg (2008)
26. Easterbrook, S., Singer, J., Storey, M.-A., Damian, D.: Selecting Empirical Methods for Software Engineering Research. In: Guide to Advanced Empirical Software Engineering, pp. 285–311 (2008)
27. Deelstra, S., Sinnema, M., Bosch, J.: Variability assessment in software product families. Information and Software Technology 51, 195–218 (2009)
28. Pfleeger, S.L., Kitchenham, B.A.: Principles of survey research: part 1: turning lemons into lemonade. SIGSOFT Softw. Eng. Notes 26, 16–18 (2001)
29. Kasunic, M.: Designing an Effective Survey, Vol. CMU/SEI-2005-HB-004 SEI, CMU (2005)
30. Etzkorn, L.H., Hughes, W.E., Davis, C.G.: Automated reusability quality analysis of OO legacy software. Information and Software Technology 43, 295–308 (2001)
31. Washizaki, H., Yamamoto, H., Fukazawa, Y.: A Metrics Suite for Measuring Reusability of Software Components. In: Proceedings of the 9th International Symposium on Software Metrics, pp. 221–225. IEEE Computer Society Press, Los Alamitos (2003)
32. Dandashi, F.: A method for assessing the reusability of object-oriented code using a validated set of automated measurements. In: Proceedings of the 2002 ACM symposium on Applied computing, pp. 997–1003. ACM, Madrid, Spain (2002)
33. Washizaki, H., Namiki, R., Fukuoka, T., Harada, Y., Watanabe, H.: A Framework for Measuring and Evaluating Program Source Code Quality. In: Münch, J., Abrahamsson, P. (eds.) PROFES 2007. LNCS, vol. 4589, pp. 284–299. Springer, Heidelberg (2007)

Test Data Generation for Event-B Models Using Genetic Algorithms

Ionut Dinca, Alin Stefanescu, Florentin Ipate, Raluca Lefticaru,
and Cristina Tudose

University of Pitesti, Department of Computer Science
Str. Targu din Vale 1, 110040 Pitesti, Romania
{name.surname}@upit.ro

Abstract. Event-B is a formal modeling language having set theory as its mathematical foundation and abstract state machines as its behavioral specifications. The language has very good tool support based on theorem proving and model checking technologies, but very little support for test generation. Motivated by industrial interest in the latter domain, this paper presents an approach based on genetic algorithms that generates test data for Event-B test paths. For that, new fitness functions adapted to the set-theoretic nature of Event-B are devised. The approach was implemented and its efficiency was proven on a carefully designed benchmark using statistically sound evaluations.

1 Introduction

Event-B [1] is a modeling language used for formal system specification and analysis. Event-B was introduced about ten years ago and it quickly attracted the attention of both academic and industrial researchers. The theoretical foundations and associated tooling were developed in several research projects, among which the most notable are two large European research projects: RODIN [2], which produced a first platform for Event-B called *Rodin*, and DEPLOY [3], which is currently enhancing this platform based on industrial feedback. Theorem-proving is the core technology within Rodin, but model-checking tools have also been developed (ProB [4]). Recently, there has been an increasing interest from the industrial partners like SAP (who belongs to DEPLOY consortium) for test generation based on Event-B models [19]. This provided the main motivation for our investigations into model-based testing using Event-B models, especially test data generation.

Model-based testing (MBT) is an approach that uses formal models as basis for automatic generation of test cases [18]. For MBT using state-based models, test generation algorithms usually traverse the state space from the initial state, guided by a certain coverage criterion (e.g. state coverage), collecting the execution paths in a test suite. Event-B models do not have an explicit state

¹ <http://rodin.cs.ncl.ac.uk> - Project running between 2004-2007

² <http://deploy-project.eu> - Project running between 2008-2012

space; instead, their state spaces are given by the value of the variables. The ProB tool [9], which is available in the Rodin platform, has a good control of the state space, being able to explore it, visualize it and verify various properties using model-checking algorithms. Such algorithms can be used to explore the state space of Event-B models using certain coverage criteria (e.g. event coverage) and thus generating test cases along the traversal. Moreover, the input data that trigger the events provides the test data associated with the test cases. Such an approach using explicit model-checking has been applied to models from the business application area by SAP [19]. The algorithms perform well for models with data with a small finite range. However, in case of variables with a large range (e.g. integers), the known state space explosion problem creates difficulties, since the model checker explores the state space by enumerating the many possible values of the variables.

This paper addresses a slightly different, but related, problem. Given a (potentially feasible) path in the Event-B model, we use meta-heuristic search algorithms (more precisely, genetic algorithms) to generate input data that trigger the execution of the path. This is a very important issue of MBT since, for models with large state spaces, paths with restrictive triggering conditions (e.g. composed conditions involving one or more = operators) are difficult to attain using the model checking approach described above. A similar problem has been addressed by recent work on search-based testing for Extended Finite State Machines (EFSMs) [8,5,20]. However, there are a number of issues that differentiate search-based testing on Event-B models from these EFSM approaches as described our position paper [16] like implicit state space, non-numerical types, non-determinism and hierarchical models. In this paper, we start addressing some of these issues, especially the non-numerical types.

The main contributions of the paper are enumerated below:

- Since the data structures used by Event-B models are predominantly set-based rather than numerical, Tracey-like [17] fitness functions for such data types are newly defined. These fitness functions are used to guide the search for the solutions in large state spaces.
- Furthermore, the encoding of non-numerical types into a chromosome is investigated. As Event-B models may use a mixture of numerical and non-numerical types, the encoding has to accommodate also such a possibility.
- The proposed search-based testing approach for Event-B is applied on a number of industry-inspired case studies. The experiments show that the approach performs better in general compared to random testing approaches.

The paper is structured as follows. We start by describing the Event-B framework in Section 2 together with a couple of representative Event-B examples in Section 3. Then we present the proposed test generation framework based on search-based techniques using genetic algorithms in Section 4. The experiments are explained in Section 5 and the conclusions are drawn in Section 6.

2 Formal Modeling with Event-B

Event-B [1] is a formal language based on the notion of abstract machines having set theory as its mathematical foundation. The two basic constructs in Event-B are *contexts* and *machines*. The contexts specify the static part of a model and contain the following elements: *carrier sets* (i.e. domains), *constants* and *axioms*. The axioms can define relations between the constants or define their domains. Machines represent the dynamic part of a model and can contain *variables*, *invariants* and *events*.

An event is composed of two main elements: *guards* which are predicates that describe the conditions that must hold for the occurrence of an event and *actions* which determine how specific variables change as a result of the event execution. An event has the following general form:

$$\text{Event} \hat{=} \mathbf{any } t \mathbf{ where } G(t, x) \mathbf{ then } S(x, t) \mathbf{ end.}$$

Above, t is a set of local parameters, x is a set of global variables appearing in the event, G is a predicate over t and x , called the guard and $S(x, t)$ represents a substitution. If the guard of an event is false, the event cannot occur and is called disabled. The substitution S modifies the values of the global variables in the set x . It can use the old values from x and the parameters from t .

For example, an event that takes a natural number parameter *value* smaller than 50 and adds it to the natural (global) variable *balance* only if *balance* is larger than 1500, can be modeled as:

$$\text{Event1} \hat{=} \mathbf{any } value \mathbf{ where } value \in \mathbb{N} \wedge value < 50 \wedge balance > 1500 \mathbf{ then} \\ balance := balance + value \mathbf{ end.}$$

Note that if the model has 10 integer variables with their range in $[1..10,000]$, then the explicit state space would have 10^{40} states, which is usually too much for a brute-force traversal algorithm of an explicit model checker. In this paper, we use meta-heuristic search techniques to deal with such large state spaces.

Let us consider another example, involving a set defined as $ITEMS = \{it1, it2, \dots, it20\}$ and an event that modifies a set variable *items*, where $items \subseteq ITEMS$ (alternatively, we can write that *items* is an element of the powerset of $ITEMS$, i.e. $items \in \mathbb{P}(ITEMS)$). We can model a situation in which the value of one global variable *oneItem* is randomly picked from the set *items* (using the Event-B operator $:\in$) and the set *items* is updated with elements from a parameter *buffer* of cardinality smaller than 5 (the cardinality is denoted by $card()$):

$$\text{Event2} \hat{=} \mathbf{any } buffer \mathbf{ where } buffer \in \mathbb{P}(ITEMS) \wedge card(buffer) < 5 \mathbf{ then} \\ oneItem :\in items \wedge items := items \cup buffer \mathbf{ end.}$$

Thus, an Event-B model is given by the defined domains, constants, variables, events that change the global variables when executed, and a set of global invariants specifying the properties required by the specification. The execution of a model starts with a special event that initializes the system, followed by the application of enabled events. At each execution step, all the guards of the events

are evaluated and the set of enabled events is computed. Then, one enabled event is non-deterministically chosen and its action is executed. The Rodin platform (<http://www.event-b.org/platform.html>), built on top of Eclipse, provides different plugins that manage and perform different tasks on the Event-B models.

3 Case Studies

In Section 5 we run the experiments on a benchmark of 5 Event-B models. The models are not industrial ones, but are inspired by industrial examples. We have been in contact with partners in the DEPLOY project that are interested in test generation from Event-B models, especially SAP, which is an industrial partner from the business software area. We have discussed a couple of MBT requirements together with a couple of sample models. For the benchmark, we made model variations such that we cover different guard and variable types.

We describe 2 out of the 5 Event-B models that we used for the benchmarks. The events of the first one contain numerical parameters, while the events of the second model focus on set parameters. The presentation of each model starts with a short description, followed by the types of the global variables (defined in the context of the Event-B model). Then, the events of the Event-B machines are listed together with their parameters. The guards and actions associated to each event are presented in a separate table.

Numerical-based model: Bank Account. The first example models a simple bank account system. The system allows the user to deposit money in the account or to withdraw money from it. The bank pays interest and charges fees. Depending on the current balance, a deposit can be in four states: overdraft, empty, silver and gold. Thus, the Event-B variables are: $balance \in \mathbb{Z}$, $transaction \in \text{BOOL}$ and $state \in \text{STATES} = \{\text{overdraft}, \text{empty}, \text{silver}, \text{gold}\}$. The machine events, whose guards and actions are given in Table 1 are the following:

- E1. *Initialization*, that initializes the bank account
- E2. *Deposit*, having the numerical parameters $amount1$ and $amount2$
- E3. *Withdraw*, having the numerical parameters $amount1$ and $amount2$
- E4. *ValidateOverdraft*
- E5. *ValidateEmpty*
- E6. *ValidateSilver*
- E7. *ValidateGold*
- E8. *PayInterest*, having the numerical parameter $value$
- E9. *ChargeFee*, having the numerical parameter fee .

Set-based model: Basket of Items. Here we model a basket of items. The system allows the user to add items, to remove items and to pick items from the basket. The system checks if the basket is empty or full or can make a special check. The global variables are: $items \in \mathbb{P}(\text{ITEMS})$, $buffer \in \mathbb{P}(\text{ITEMS})$, $isEmpty \in \text{BOOL}$, $isFull \in \text{BOOL}$, $CAPACITY \in \mathbb{N}$, and $count \in \mathbb{N}$ with the invariants $count \geq 0 \wedge count \leq CAPACITY$ and $count = \text{card}(items)$, where $\text{ITEMS} = \{it1, it2, \dots, it20\}$. The Event-B events, whose guards and actions are given in Table 2 are the following:

Table 1. Guards and actions of Bank Account events

Ev Guards	Actions
E1: $TRUE$	$balance := 0, state := empty$ $transaction := FALSE$
E2: $amount1 + amount2 > 200 \wedge$ $amount1 \in \mathbb{N} \wedge amount2 \in \mathbb{N}$	$balance := balance + amount1 + amount2$ $transaction := TRUE$
E3: $balance > 0 \wedge amount1 + amount2 < 1000 \wedge$ $balance - amount1 - amount2 > -100 \wedge$ $amount1 \in \mathbb{N} \wedge amount2 \in \mathbb{N}$	$balance := balance - amount1 - amount2$ $transaction := TRUE$
E4: $balance < 0 \wedge balance > -100$	$state := overdraft$
E5: $balance = 0$	$state := empty$
E6: $balance > 0 \wedge balance < 1000$	$state := silver$
E7: $balance \geq 1000$	$state := gold$
E8: $balance > 1500 \wedge$ $value \leq 50 \wedge value > 0 \wedge value \in \mathbb{N}$	$balance := balance + value$
E9: $fee > 0 \wedge fee < 50 \wedge fee \in \mathbb{N} \wedge$ $transaction = TRUE$	$balance := balance - fee$ $transaction := FALSE$

- E1. *Initialization*, that initializes the basket of items
E2. *PickItems*, with the set parameter *its*
E3. *AddItems*
E4. *RemoveItems*
E5. *ValidateEmpty*
E6. *CheckSpecial*
E7. *ValidateFull*.

4 Test Data Generation for Event-B Models Using Genetic Algorithms

Before describing our test generation approach, let us establish the problem to be solved. First, let us note that Event-B specifications are event-based rather than state-based models. Formally, these are abstract state machines [4] in which the (implicit) states are given by the (global) values of the *variables* on which the events operate. Each event is given by a triplet consisting of (1) the *parameters* (local variables) used by the event, (2) the guards which constrain the event application (the guards may involve both local and global variables) and (3) the actions of event, which may change the values of the global variables. The events produce the transitions between states: the guards establish the valid source state(s) of the transition while the actions produce the target state(s). In general, the application of an event depends on the values of the parameters it receives. If we want to execute a path (sequence of events) through the model, we will need to find appropriate parameter values for each event in the sequence

Table 2. Guards and actions of Basket of Items events

Ev Guards	Actions
E1: $TRUE$	$items := \emptyset, buffer := \emptyset, count := 0$ $CAPACITY := card(ITEMS)$ $isEmpty := TRUE, isFull := FALSE$
E2: $its \subseteq ITEMS$	$buffer := its$
E3: $buffer \subseteq ITEMS \wedge card(buffer) > 5 \wedge$ $card(buffer) + count \leq CAPACITY$	$items := items \cup buffer$ $count := card(items \cup buffer)$ $isEmpty := FALSE$
E4: $buffer \subseteq items \wedge card(buffer) > 3 \wedge$ $count - card(buffer) \geq 0$	$items := items \setminus buffer$ $count := card(items \setminus buffer)$
E5: $items = \emptyset \wedge count = 0$	$isEmpty := TRUE$
E6: $\{it1, it20\} \subseteq items \wedge card(items) < 6$	$buffer := items$
E7: $count = CAPACITY$	$isFull := TRUE$ $items := \emptyset, count := 0$

(i.e. which satisfy the corresponding guards). This is the problem we will solve using a genetic algorithm. Naturally, the prerequisite is that a set of paths, which satisfies the given test requirement has already been found.

In general, this requirement is expressed as a level of coverage of the model. Various levels of coverage for finite state machines exist in the literature [3,18] and some can be adapted to Event-B models without the need to transform the model into an explicit state machine (for large systems this transformation may be impractical). For example, transition coverage for a finite state machine requires every transition to be triggered at least once. Similarly, for Event-B models, event coverage will involve the execution of every event at least once. This type of coverage can be generalized by requiring that each feasible sequences of events of a given length k is executed at least once. Obviously, in order to decide if a path is feasible or not it may be necessary to effectively find test data (parameter values) which triggers it. Consequently, the potentially feasible paths can be selected first by deleting paths which contain obvious contradictory constraints (e.g. both C and $\neg C$) and then the test data generation algorithm is applied to each such path. Other types of coverage may also be defined but this beyond the scope of this paper.

In this paper, we assume that we have a set of paths (that cover, for instance, all events of the model). For each path of the given set, we seek appropriate test data, i.e. event parameters which enable the events in the path. It may be possible that the test data for the selected path has not been found, either because of the complexity of the guard constraints or simply because the path is infeasible; if this is the case, a new path is selected. Note that the paper does not address the issue of path selection, but only the test generation for the chosen path(s).

Below we present the theoretical instruments based on genetic algorithms for the above problem. First, Subsection 4.1 provides the background on genetic algorithms. Then, the Subsections 4.2 and 4.3 describe the main ingredients of the approach, i.e. the encoding of the sought solutions into chromosomes and the fitness function that guides the search into the solution space, respectively.

Note that among the different meta-heuristic algorithms, for convenience, in this paper we have chosen to use the class of genetic algorithms [13], because they are widely used in search-based testing approaches and have good tooling support. However, we plan in the future to experiment with other types of algorithms like simulated annealing or particle swarm optimization.

4.1 Genetic Algorithms

Genetic algorithms (GAs) [13] are a particular class of *evolutionary algorithms*, that use techniques inspired from biology, such as selection, recombination (crossover) and mutation. GAs are used for problems which cannot be solved using traditional techniques and for which an exhaustive search of the solution space is impractical. In particular, the application of GAs to the difficult problem of test data generation recently received an increased attention from the testing research community [11,10].

GAs basic approach is to encode a population of potential solutions on some data structures, called *chromosomes* (or *individuals*) and applying recombination and mutation operators to these structures. A high-level description of a genetic algorithm [10,13] is given in Fig. 1. The *fitness (or objective) function* assigns a score (called fitness) to each chromosome in the current population. The fitness of a chromosome depends on how close that chromosome is to the solution of the problem. Throughout this paper, the fitness is considered to be positive and finding a solution corresponds to minimizing the fitness function, i.e. a solution will be a chromosome with fitness 0. The algorithm terminates when some stopping criterion has been met, for example when a solution is found, or when the number of generations has reached the maximum allowed limit.

Various mechanisms for selecting the individuals to be used to create offspring, based on their fitness, have been devised [6]. GA researchers have experimented with mechanisms such as sigma scaling, elitism, Boltzmann selection, tournament, rank and steady-state selection [13].

After the selection step, recombination takes place to form the next generation from parents and offspring. The mutation operator is then applied. These two operations, crossover and mutation, depend on the type of encoding used and so they are discussed in more detail in the next subsection.

4.2 Chromosome Encodings

Consider a path $event_1 \dots event_n$ in the Event-B model. A *chromosome* (possible solution) is a list of values, $x = (x_1, \dots, x_m)$ for the event parameters of the path events (in the order they appear). More formally, if p_{i1}, \dots, p_{ik_i} are the parameters of $event_i, 1 \leq i \leq n$, then x represents a list of values for parameters

```

Randomly generate or seed initial population  $P$ 
Repeat
  Evaluate fitness of each individual in  $P$ 
  Select  $P'$  from  $P$  according to selection mechanism
  Recombine parents from  $P'$  to form new offspring
  Construct  $P'$  from parents and offspring
  Mutate  $P'$ 
   $P \leftarrow P'$ 
Until Stopping Condition Reached

```

Fig. 1. Genetic Algorithm

$p_{11} \dots p_{nk_n}$. Naturally, $m = k_1 + \dots + k_n$ can differ from the number n of events in the sequence. If the values x satisfy all guards and, consequently, trigger the path, then x is a solution for the given path. For numerical data, the chromosomes are integer-encoded, each gene representing one parameter.

Consider, for example, the path $E2\ E8\ E9\ E3\ E7$ from the Bank Account example presented earlier (technically, any path of a model starts with the special event *Initialization* ($E1$), but for simplicity when we mention the events of a path we skip $E1$). There are five events in the path: $E2$ (*Deposit*), which receives *amount1* and *amount2* as parameters, $E8$ (*PayInterest*), with parameter *value*, $E9$ (*ChargeFee*), with parameter *fee*, $E3$ (*Withdraw*), with parameters *amount1* and *amount2* and $E7$ (*ValidateGold*), with no parameters. Since all 6 parameters have numerical types, a chromosome for the above path will be a list of 6 integers.

An additional problem occurs when non-numerical types are involved since such values will have to be encoded into the chromosome. The applications we have considered use enumeration types as well as types derived from these using traditional set operators (\cup , \setminus , \times). For a k -valued type $T = \{v_1, \dots, v_k\}$, a set parameter S which is a subset of T , i.e. $S \subseteq T$, is represented by a bitmap of length k , which has 1 on the i th position in the bitmap if $v_i \in S$, and 0 otherwise. Then, a chromosome corresponding to parameters $p_1 \dots p_m$ will be a list of values $x_1 \dots x_m$, in which each value is encoded as appropriate. The applications we have considered use both numerical and non-numerical types and so some values in the chromosome are represented by simple integers whereas other values are encoded as bitmaps. Once we generated a population of chromosomes, the operations of crossover and mutation are applied as described below.

Crossover. For mixed chromosomes (with both binary and integer genes) and binary-only chromosomes, a *single-point crossover* is used. This randomly chooses a locus and exchanges the subsequences before and after that locus between two chromosomes to create two new offspring. For example, the strings 00000000 and 11111111 could be crossed over at the third locus to produce the two offspring 00011111 and 11100000. The crossover is applied to individuals selected at random, with a probability (rate) p_c . Depending on this rate, the next generation will contain the parents or the offspring.

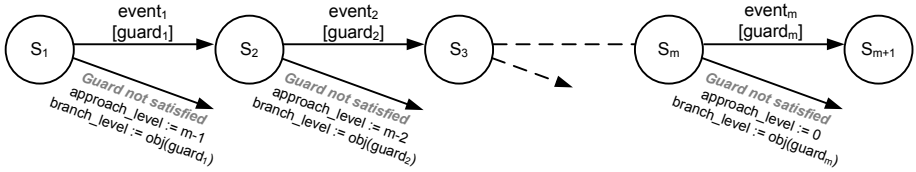


Fig. 2. Calculating the fitness function

For integer chromosomes, we used a *heuristic real value crossover*, inspired from [12], that showed to be the most efficient type of crossover for our problem. This uses the fitness of the individual for determining the direction of the search. For parents $x = (x_1, \dots, x_m)$, $y = (y_1, \dots, y_m)$, x fitter than y , one offspring $z = (z_1, \dots, z_m)$ is generated, with z_i being the integer-rounded value of $\alpha \cdot (x_i - y_i) + x_i$, $\alpha \in (0, 1)$. Heuristic real value and single-point crossovers can be combined.

Mutation is used to introduce variation in the population by randomly changing genes in the chromosome. Mutation can occur at each bit position in a string with some probability p_m , usually very small [13]. For binary genes, the mutation operator randomly flips some bits in a chromosome. For example, the string 00000100 could be mutated in its second position to yield 01000100. For integer genes, the gene value is replaced by another integer value that is randomly chosen from the same interval.

4.3 Fitness Function

The algorithm evaluates a candidate solution by executing each event with the values encoded in the chromosome's genes until the guard of the current event is not satisfied. The fitter individuals are the ones which enable more events from the given path. They are rewarded with a lower fitness value. The fitness function is calculated using a formula widely used in the search-based testing literature [11, 8], using two components. The first evaluates how close a chromosome is to executing the given path, by counting the events executed. The second measures how close is the first unsatisfied guard predicate to being true.

$$fitness := approach_level + normalized_branch_level$$

The first component, *approach (approximation) level* is similar a metric in evolutionary structural test data generation [11]. This is calculated by $m - 1 - n$, where m is the length of the path to be executed and n is the number of events successfully executed until the first unsatisfied guard on the path, as in Fig. 2.

A fitness function formed only from the approach level has many plateaux (i.e. for each value $0, 1, \dots, m - 1$) and it would not offer enough guidance to the search. Consequently, the second component, called *branch level*, was introduced. This computes, for the place where the actual path diverges from the required one, how close was the guard predicate to being true.

Table 3. Tracey’s objective functions for relational predicates and logical connectives. The value K , $K > 0$, refers to a constant which is always added if the term is not true.

Relational predicate or logical connective	Objective function obj
$a = b$	if $abs(a - b) = 0$ then 0 else $abs(a - b) + K$
$a \neq b$	if $abs(a - b) \neq 0$ then 0 else K
$a < b$	if $a - b < 0$ then 0 else $(a - b) + K$
$a \leq b$	if $a - b \leq 0$ then 0 else $(a - b) + K$
$a > b$	if $b - a < 0$ then 0 else $(b - a) + K$
$a \geq b$	if $b - a \leq 0$ then 0 else $(b - a) + K$
Boolean	if $TRUE$ then 0 else K
$a \wedge b$	$obj(a) + obj(b)$
$a \vee b$	$min(obj(a), obj(b))$
$a \text{ xor } b$	$obj((a \wedge \neg b) \vee (\neg a \wedge b))$
$\neg a$	Negation is moved inwards and propagated over a

For numeric types, the *branch level* can be derived from the guards predicates using Tracey’s objective functions as shown in Table 3 [17,10]. The *branch level* is then mapped onto the interval $[0,1)$ or normalized.

We extended the calculation of the *branch level* to applications which involve set theory based constraints as described below. The applications considered use basic types that can be mapped onto either an interval $([p..q], 0 \leq p < q)$ or an enumeration of non-negative integers $(\{p_1, \dots, p_n\}, n \geq 1, p_i \geq 0, 1 \leq i \leq n)$. Furthermore, the derived types use the \cup , \cap , \setminus and \times set operators. Then the objective function for the $a \in A$ and $a \notin A$ predicates can be derived using the transformations given at the top of Table 4. The formulae are then extended for the \subseteq and $=$ set operators, as shown at the bottom of Table 4.

5 Experiments

We have implemented our approach as a plugin for the Eclipse-based Rodin platform for Event-B. The plugin is designed to automatically generate test data for given paths in the Event-B model. It can generate test data, i.e. the input parameters for the events on the given path, employing the fitness function described in Section 4.3. The execution of the events (including the initialization) was performed using the Event-B model simulation of the ProB plugin [9].

For the benchmark of 5 models mentioned in Section 3, we have considered a set of 18 random paths likely to be feasible, which covered all the events from the models. The paths length varied between 2 and 5 events (without counting the initialization event). The number of parameters on each path varied between: (a) 1 and 7 for numerical models; (b) 1 and 2 non-numerical parameters, such as $x \in \mathbb{P}(ITEMS)$ for set examples; (c) 2 – 4 set parameters and 2 – 3 numerical

Table 4. The extension of Tracey's objective functions to set operators

Predicate involving \in for basic or derived sets	Objective function obj
$a \in [p, q]$	$obj((a \geq p) \wedge (a \leq q))$
$a \notin [p, q]$	$obj((a < p) \vee (a > q))$
$a \in \{p_1, \dots, p_n\}$	$obj((a = p_1) \vee \dots \vee (a = p_n))$
$a \notin \{p_1, \dots, p_n\}$	$obj((a \neq p_1) \wedge \dots \wedge (a \neq p_n))$
$a \in A \cup B$	$obj((a \in A) \vee (a \in B))$
$a \in A \cap B$	$obj((a \in A) \wedge (a \in B))$
$a \in A \setminus B$	$obj((a \in A) \wedge (a \notin B))$
$(a, b) \in (A, B)$	$obj((a \in A) \wedge (b \in B))$
Predicates for \subseteq and $=$ operators	Objective function obj
$[p, q] \subseteq A$	$obj(\bigwedge_{i=p}^q (i \in A))$
$\{p_1, \dots, p_n\} \subseteq A$	$obj((p_1 \in A) \wedge \dots \wedge (p_n \in A))$
$[p, q] \not\subseteq A$	$obj(\bigvee_{i=p}^q (i \notin A))$
$\{p_1, \dots, p_n\} \not\subseteq A$	$obj((p_1 \notin A) \vee \dots \vee (p_n \notin A))$
$A = B$	$obj((A \subseteq B) \wedge (B \subseteq A))$
$A \neq B$	$obj((A \not\subseteq B) \vee (B \not\subseteq A))$

parameters for the mixed model. The codification used was: integer-valued for numerical parameters (the integer range was fixed to 2000) and bitmap for set parameters.

As recommended in [2], a search algorithm (GA in this case) should be compared with random search in order to check that the algorithm is not simply successful because the search problem is easy. Therefore, we tried to generate test data for the 18 selected paths mentioned above, denoted by $P1 - P18$, using the two methods: *search-based testing with genetic algorithms (GA)* and *random testing (RT)*. For each path and each test generation method, 30 runs were performed (this number was also recommended in [2]). A run is considered *successful* if it can produce input test data that can trigger the whole path, or equivalently, the fitness function associated has the value 0. The run ends when a solution was found or when the maximum number of generations was reached.

Using genetic algorithms, the amount of time needed to obtain test data for a path, i.e. the actual values of the parameters which trigger the path, varied between 1 second (for very simple paths, where the solution could be found from the first generation) and 60 seconds (for complex paths).

The genetic algorithm framework used for experimentation was the open source Java Genetic Algorithms Package (JGAP) [7]. The maximum number of generations for the genetic algorithm was set to 100 and the population size to 20. The selection operator employed was *BestChromosomesSelector*, an elitist operator, the mutation rate was $p_m = 1/10$ and the crossover was single-point (for non-numerical parameters) or heuristic crossover (for numerical ones), as presented in Section 4.2.

For random testing, the same library was used: instead of applying recombination or mutation, the population was randomly generated at each step, ensuring this way an equal treatment, i.e. an equal number of generations (or fitness function evaluations) for both methods, GA and RT. For each run, the generation when the solution was found was recorded and Table 5 presents the summarizing data: the success rate for each method (percent of successful runs from the 30 ones considered) and other descriptive statistics, e.g. the average (mean) number of generations, the median and the standard deviation.

Statistical tests should be realized to support the comparison of GA and RT runs. In our experiments we have used two statistical tests: the parametric t -test and the non-parametric Mann-Whitney U-test. The null hypothesis (H_0) is thus formulated as follows: *There is no difference in efficiency (the number of generations needed to find a solution) between GA and RT.* The alternative hypothesis (H_a) follows: there is a difference between the two approaches, GA and RT. The two tests measure different aspects: the t -test measures the difference in mean values (the null hypothesis is $H_0 : \mu_1 = \mu_2$), whereas the Mann-Whitney U-test measures their difference in medians ($H_0 : \theta_1 = \theta_2$), i.e. whether the observations in one data sample are more likely to be larger than observations in the other sample.

The test results and the p -values obtained are given in Table 5. In the columns t -test and U-test, the sign ‘+’ stands for rejecting the null hypothesis (consequently, there is a statistically significant difference between GA and RT results), while the ‘-’ indicates that the null hypothesis cannot be rejected at the significance level considered, $\alpha = 0.01$. The p -value computed by the statistical test is also provided, excepting the case when it can not be computed, e.g. when both approaches were able to find a solution from the first generation for all the runs (paths $P16, P17$), where ‘+’ stands for not computed.

Some standardized effect size measures were also used and they are given in the last two columns: the Vargha and Delaney’s A statistic, the Cohen’s D coefficient. The Vargha and Delaney’s A statistic [2] is a performance measure, used to quantify the probability that GA yield ‘better values’ than RT. In our case, ‘better values’ means lower number of generations needed to obtain a solution.

The Vargha and Delaney’s statistics is given in the column ‘A’. For simple paths, where RT and GA provide the solution in the same number of generations, the effect size is 0.5. For more complex paths, a value of 0.82 means that we would obtain better results in 82% of the time with GA (they guide the search to success in a lower number of generations). It is worth noting that *GA clearly outperformed RT for 14 out of 18 paths considered*, and the difference in terms of success rate, average (or median) number of generations was significant.

The last column of Table 5 presents the Cohen’s D coefficient, which is computed as the absolute difference between two means, divided by a pooled standard deviation of the data [2,15]. According to [15], Cohen has proposed the following ‘D values’ as criteria for identifying the magnitude of an effect size:

Table 5. Success rates, results of the statistical tests and effect size measures

Path	Meth.	Success rate	Avg. gen.	Median	Std. dev.	t-test p-val	U-test p-val	A	D
P1	GA	100.0%	18.2	12.0	18.8	+	+	1.00	5.85
P1	RT	3.3%	99.0	100.0	5.3	< 0.001	< 0.001		
P2	GA	100.0%	14.9	11.0	12.5	+	+	1.00	6.45
P2	RT	3.3%	97.6	100.0	13.1	< 0.001	< 0.001		
P3	GA	96.7%	22.7	14.5	19.8	+	+	0.98	4.61
P3	RT	10.0%	96.9	100.0	11.1	< 0.001	< 0.001		
P4	GA	100.0%	12.4	8.0	11.6	+	+	0.82	1.17
P4	RT	96.7%	35.0	32.5	24.8	< 0.001	< 0.001		
P5	GA	66.7%	53.3	44.5	38.7	+	+	0.83	1.71
P5	RT	0.0%	100.0	100.0	0.0	< 0.001	< 0.001		
P6	GA	100.0%	23.5	21.0	9.2	+	+	1.00	11.73
P6	RT	0.0%	100.0	100.0	0.0	< 0.001	< 0.001		
P7	GA	100.0%	12.7	12.0	4.5	+	+	1.00	16.72
P7	RT	6.7%	98.5	100.0	5.7	< 0.001	< 0.001		
P8	GA	100.0%	16.9	17.0	4.4	+	+	1.00	26.59
P8	RT	0.0%	100.0	100.0	0.0	< 0.001	< 0.001		
P9	GA	100.0%	13.7	13.0	2.4	+	+	1.00	12.46
P9	RT	3.3%	98.3	100.0	9.3	< 0.001	< 0.001		
P10	GA	100.0%	30.9	31.5	6.3	+	+	1.00	15.51
P10	RT	0.0%	100.0	100.0	0.0	< 0.001	< 0.001		
P11	GA	96.7%	20.0	13.0	22.6	+	+	0.98	4.64
P11	RT	3.3%	98.6	100.0	7.9	< 0.001	< 0.001		
P12	GA	100.0%	13.5	13.0	2.8	+	+	1.00	43.66
P12	RT	0.0%	100.0	100.0	0.0	< 0.001	< 0.001		
P13	GA	100.0%	11.9	11.5	2.2	+	+	1.00	56.56
P13	RT	0.0%	100.0	100.0	0.0	< 0.001	< 0.001		
P14	GA	100.0%	1.3	1.0	1.6	-	-	0.47	0.29
P14	RT	100.0%	1.0	1.0	0.0	0.28	0.49		
P15	GA	100.0%	1.0	1.0	0.0	-	-	0.50	†
P15	RT	100.0%	1.0	1.0	0.0	†	1.00		
P16	GA	100.0%	1.0	1.0	0.0	-	-	0.50	†
P16	RT	100.0%	1.0	1.0	0.0	†	1.00		
P17	GA	90.0%	16.9	5.5	29.9	+	+	0.91	1.79
P17	RT	53.3%	72.2	83.5	31.7	< 0.001	< 0.001		
P18	GA	100.0%	1.8	1.0	2.4	-	-	0.53	0.17
P18	RT	100.0%	1.5	1.0	0.8	0.53	0.63		

a) small effect size: $D \in (0.2, 0.5)$, b) medium effect size $D \in [0.5, 0.8)$, c) large effect size $D \in [0.8, \infty)$. According to this classification, it can be easily noticed that the *difference between the results obtained with GA versus RT correspond for most paths (14 out of 18) to a large effect size.*

6 Final Discussion

Bottom line. In this paper, we have presented an approach based on genetic algorithms that allows generating test data for event paths in the Event-B framework. One distinguishing feature of Event-B is its set-theoretic foundation, meaning that in Event-B models, numerical variables are used together with non-numerical types based on sets. To address this, we extended the fitness functions available in the search-based testing literature to set types. Moreover, the encoding of the sought solutions included mixed chromosomes containing both numerical and non-numerical types. Finally, we followed standard statistical guidelines [2] to demonstrate the efficiency and effectiveness of our implementation on a diversified benchmark inspired by discussions with the industry.

Related work. The only approach of test generation for Event-B models is based on explicit model-checking [19] with ProB [9], which suffers from the classical state space explosion problem. There is also related work on applying search-based techniques to EFSMs [8,5,20]. Differently from these, we address a different modeling language and tackle non-numerical types. However, we can certainly extend our work with ideas from these papers, e.g. regarding feasible path generation, or from previous work on test generation from B models (the precursor of Event-B language, even though B is not an event-based language) [14, [18, ch.3].

Future work. Since the goal is to develop a test method that scales for industrial Event-B models, we have performed a survey of 29 publicly available Event-B models posted by the DEPLOY academic and industrial partners³. Beside the large size of industrial models, there are a couple of other dimensions still to be addressed. For instance, Event-B uses a rich set of operations as well as complex data based on set relations, sets of sets or partial functions. In principle, these can be mapped to sets and use the proposed methods but this may not scale, so the fitness functions and encodings might need to be further specialized for these operators. Moreover, industrial models are usually decomposed in order to mitigate modeling complexity, which means that we have to extend our methods to work for modular and component-based models.

Acknowledgment This work was partially supported by project DEPLOY (EC-grant no. 214158) and Romanian Research Grant CNCS-UEFISCDI no. 7/05.08.2010 at the University of Pitesti. We thank V. Kozyura and S. Wiczorek from SAP Research for interesting models and inspiring discussions.

³ <http://deploy-eprints.ecs.soton.ac.uk/view/type/rodin=5Farchive.html>

References

1. Abrial, J.-R.: *Modeling in Event-B - System and Software Engineering*. Cambridge University Press, Cambridge (2010)
2. Arcuri, A., Briand, L.: A practical guide for using statistical tests to assess randomized algorithms in software engineering. In: *Proc. ICSE* (to appear, 2011)
3. Binder, R.V.: *Testing Object-Oriented Systems: Models, Patterns, and Tools*, Object Technology. Addison-Wesley, London (1999)
4. Börger, E., Stärk, R.: *Abstract State Machines: A Method for High-Level System Design and Analysis*. Springer, Heidelberg (2003)
5. Derderian, K., Hierons, R.M., Harman, M., Guo, Q.: Estimating the feasibility of transition paths in extended finite state machines. *Autom. Softw. Eng.* 17(1), 33–56 (2010)
6. Goldberg, D.E., Deb, K.: A comparative analysis of selection schemes used in genetic algorithms. In: *FOGA*, pp. 69–93 (1990)
7. Meffert, K., et al.: JGAP - Java Genetic Algorithms and Genetic Programming Package, <http://jgap.sf.net> (last visited March 2011)
8. Lefticaru, R., Ipate, F.: Functional search-based testing from state machines. In: *Proc. ICST 2008*, pp. 525–528. IEEE Computer Society Press, Los Alamitos (2008)
9. Leuschel, M., Butler, M.J.: ProB: an automated analysis toolset for the B method. *Int. J. Softw. Tools Technol. Transf.* 10(2), 185–203 (2008)
10. McMinn, P.: Search-based software test data generation: A survey. *Softw. Test. Verif. Reliab.* 14(2), 105–156 (2004)
11. McMinn, P., Holcombe, M.: Evolutionary testing of state-based programs. In: *GECCO 2005: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pp. 1013–1020. ACM, New York (2005)
12. Michalewicz, Z.: *Genetic algorithms + data structures = evolution programs* (3rd ed.). Springer, London (1996)
13. Mitchell, M.: *An Introduction to Genetic Algorithms*. MIT Press, Cambridge (1998)
14. Satpathy, M., Butler, M., Leuschel, M., Ramesh, S.: Automatic Testing from Formal Specifications. In: Gurevich, Y., Meyer, B. (eds.) *TAP 2007*. LNCS, vol. 4454, pp. 95–113. Springer, Heidelberg (2007)
15. Sheskin, D.J.: *Handbook of Parametric and Nonparametric Statistical Procedures*, 4th edn. Chapman & Hall/CRC (2007)
16. Stefanescu, A., Ipate, F., Lefticaru, R., Tudose, C.: Towards search-based testing for Event-B models. In: *Proc. of 4th Workshop on Search-Based Software Testing (SBST 2011)*. IEEE, Los Alamitos (to appear, 2011)
17. Tracey, N.J.: *A Search-based Automated Test-Data Generation Framework for Safety-Critical Software*. PhD thesis, University of York (2000)
18. Utting, M., Legear, B.: *Practical Model-Based Testing: A Tools Approach*. Morgan Kaufmann Publishers Inc, San Francisco (2006)
19. Wiczorek, S., Kozyura, V., Roth, A., Leuschel, M., Bendisposto, J., Plagge, D., Schieferdecker, I.: Applying Model Checking to Generate Model-Based Integration Tests from Choreography Models. In: Núñez, M., Baker, P., Merayo, M.G. (eds.) *TESTCOM 2009*. LNCS, vol. 5826, pp. 179–194. Springer, Heidelberg (2009)
20. Yano, T., Martins, E., de Sousa, F.L.: Generating feasible test paths from an executable model using a multi-objective approach. In: *Proc. ICSTW 2010*, pp. 236–239. IEEE Computer Society, Los Alamitos (2010)

A Parallel Tree Based Strategy for T-Way Combinatorial Interaction Testing

Mohammad F.J. Klaib¹, Sangeetha Muthuraman², and A. Noraziah²

¹ Department of Software Engineering, Faculty of Science and Information Technology,
Jadara University, Irbid, Jordan

² Faculty of Computer Systems and Software Engineering, University Malaysia Pahang,
Pahang, Malaysia

mohammadklaib@jadara.edu.jo, noraziah@ump.edu.my

Abstract. All software systems are built with basic components which interact with each other through predefined combination rules. As the number of components increases, the interactions between the components also increases exponentially which cause the combinatorial explosion problem. This mean complete (exhaustive) testing becomes unreasonable due to the huge number of possible combinations. Although 2-way interaction testing (i.e. pairwise testing) can relief and detect 50-97 percent of errors, empirical evidence has proved that 2-way interaction testing is a poor strategy for testing highly interactive systems and it has been showed that most of the errors are triggered by the interaction of 2-6 input parameters. In this paper we enhanced our previous strategy, “A Tree Based Strategy for Test Data Generation and Cost Calculation” by applying parallel algorithms to go beyond pairwise testing. The proposed strategy can support higher interaction testing. The designed algorithms are described in details with efficient empirical results.

Keywords: parallel algorithms, combinatorial interaction testing, software testing, T-way testing.

1 Introduction

Testing [1] is an activity that aims to evaluate the attributes or capabilities of software or hardware products, and determines if the products have met their requirements. Testing in general is a very important phase of the development cycle for both software and hardware products [2], [3]. Testing helps to reveal the hidden problems in the product, which otherwise goes unnoticed providing a false sense of well being. It is said to cover 40 to 50 percent of the development cost and resources [7]. Although important to quality and widely deployed by programmers and testers, testing still remains an art. A good set of test data is one that has a high chance of uncovering previously unknown errors at a faster pace. For a successful test run of a system, we need to construct a good set of test data covering all interactions among system components.

Failures of hardware and software systems are often caused due to unexpected interactions among system components [24]. The failure of any system may be catastrophic that we may lose very important data or fortunes or sometimes even lives. The main reason for failure is the lack of proper testing. A complete test requires testing all possible combinations of interactions, which can be exorbitant even for medium sized projects due to the huge number of combinations (Combinatorial explosion problem).

Testing all pairwise (2-way) interactions between input components helps to reveal the Combinatorial explosion problem and can ensure the detection of 50 – 97 percent of faults [10], [11], [12], [13], [14], [15], [16], [23]. Although using pairwise testing gives a good percentage of reduction in fault coverage, empirical studies show that pairwise testing is not sufficient enough for highly interactive systems [9], [17], [4] and constructing a minimum test set for combinatorial interaction is still a NP complete problem [14], [7] and there is no strategy can claim that it has the best generated test suite size for all cases and systems. Therefore, based on the above argument, this work comes to extend our previous strategy “A Tree Based Strategy for Test Data Generation and Cost Calculation” by applying parallel algorithms to go beyond pairwise testing (2-way interactions). The proposed strategy can support higher interaction testing.

The remainder of this paper is organized as follows. Section 2 presents the related work. In Section 3, the proposed tree generation and the iterative T-way cost calculation strategy is illustrated and its correctness has been proved with an example. Section 4 proves the performance of the proposed strategy with efficient empirical results. Section 5 provides the conclusion and lists the main advantages of the proposed strategy.

2 Related Work

Most of existing strategies support pairwise combinatorial interaction testing and a few have been extended to work for T-way testing.

AETG [10],[15] and its variant mAETG [21] employ the computational approach based on the criteria that every test case covers as many uncovered combinations as possible. The AETG uses a random search algorithm and hence the test cases are generated in a highly non-deterministic fashion [22].

In Genetic algorithm [14] an initial population of individuals (test cases) are created and then the fitness of the created individuals is calculated. This approach follows a non deterministic methodology similar to the Ant Colony Algorithm [14] in which each path from start to end point is associated with a candidate solution.

IPO [16] Strategy for pairwise testing starts constructing the test cases by considering the first two parameters, then uses a horizontal and vertical growth until all the pairs in the covering array are covered in a deterministic fashion. IPOG [9], [17] strategy extends IPO to support T-way interactions.

The IRPS Strategy [23] linked lists to search best test cases in a deterministic fashion. G2Way [8], [7] uses backtracking strategy to generate the test cases. TConfig

[18] uses recursive algorithm for T-way testing by applying the theory of orthogonal Latin squares. Jenny [19] first covers one way interaction, then pairs of features, then triples, and so forth up to the n-tuples requested by the user. WHITCH is IBM's Intelligent Test Case Handler [5], [6]. With the given coverage properties it uses combinatorial algorithms to construct test suites over large parameter spaces. TVG [20] combines both behaviour and data modelling techniques.

Although the importance work that have been done in the past by researchers, test suite generation for combinatorial interaction testing still remains a research area and NP complete problem that needs more exploration.

3 The Proposed Strategy

The proposed strategy constructs in parallel the testing tree based on the number of parameters and values. Number of base branches depends on the number of values of the first parameter. i.e. if the first parameter has 3 values then the tree also would have 3 base branches. Therefore every branch construction starts by getting one value of the first parameter i.e. branch T1 gets the first value, T2 gets the second value and so on. After the base branches are constructed one child thread is assigned to every branch and the further construction takes place in a parallel manner. Each of the branches considers all values of all the other parameters two, three...T where T is the total number of parameters. All the branches consider the values of the parameters in the same order. Suppose the following simple system with four parameters to illustrate the concept of the algorithm:

- Parameter A has two values A1 and A2.
- Parameter B has one value B1.
- Parameter C has three values C1, C2 and C3.
- Parameter D has two values D1 and D2.

Here the illustration will be for a 3-way combinatorial interactions testing. The algorithm starts constructing the test-tree by considering the first parameter. As the first parameter has two values the tree is said to have two base branches with the first branch using A1 and the second branch using A2. Then each of the branches is constructed in parallel by considering all the values of the second parameter, then the third and fourth and so on. When the branches are fully constructed the leaf nodes gives all the test cases that has to be considered for cost calculation. Since all of the branches are constructed in parallel there is a significant reduction in time. Figure 1 shows the test tree for the system above.

Figure 1 below shows how the test-tree would be constructed. The initial test cases are T1 (A1, B1, C1, D1), T2 (A1, B1, C1, D2), T3 (A1, B1, C2, D1), T4 (A1, B1, C2, D2), T5 (A1, B1, C3, D1), T6 (A1, B1, C3, D2), T7 (A2, B1, C1, D1), T8 (A2, B1, C1, D2), T9 (A2, B1, C2, D1), T10 (A2, B1, C2, D2), T11 (A2, B1, C3, D1) and T12 (A2, B1, C3, D2).

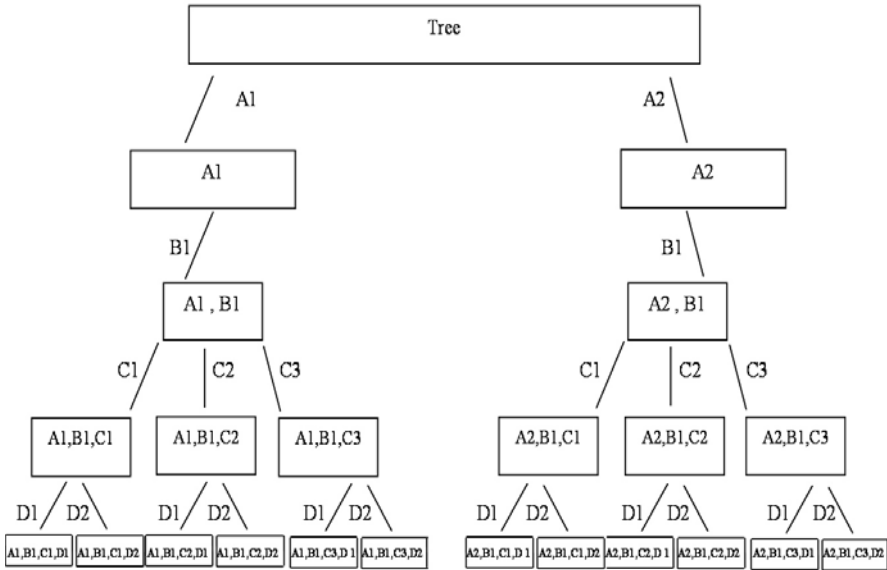


Fig. 1. Test-tree construction

Once the parallel tree construction is over we are ready with all the test cases to start the parallel iterative cost calculation. In this strategy the cost of the leaf nodes in each of the lists are calculated in parallel in order to reduce the execution time.

Table 1. 3-way interaction covering array

A, B, C	A, B, D	A, C, D	B, C, D
<i>A1, B1, C1</i>	<i>A1, B1, D1</i>	<i>A1, C1, D1</i>	<i>B1, C1, D1</i>
<i>A1, B1, C2</i>	<i>A1, B1, D2</i>	<i>A1, C1, D2</i>	<i>B1, C1, D2</i>
<i>A1, B1, C3</i>	<i>A2, B1, D1</i>	<i>A1, C2, D1</i>	<i>B1, C2, D1</i>
<i>A2, B1, C1</i>	<i>A2, B1, D2</i>	<i>A1, C2, D2</i>	<i>B1, C2, D2</i>
<i>A2, B1, C2</i>		<i>A1, C3, D1</i>	<i>B1, C3, D1</i>
<i>A2, B1, C3</i>		<i>A1, C3, D2</i>	<i>B1, C3, D2</i>
		<i>A2, C1, D1</i>	
		<i>A2, C1, D2</i>	
		<i>A2, C2, D1</i>	
		<i>A2, C2, D2</i>	
		<i>A2, C3, D1</i>	
		<i>A2, C3, D2</i>	

The cost of a particular test case is the maximum number of T-way combinations that it can cover from the covering array. Table 1 shows the covering array for 3-way combination i.e. [A, B, C], [A, B, D], [A, C, D] and [B, C, D], for the example in Figure 1. The covering array for the above example has 28 3-way interactions which have to be covered by any test suite generated to enable a complete 3-way interaction testing of the system. Table 2 shows how the cost calculation works iteratively to generate the test suite. Table 2 also shows the order in which the various test cases are actually included in the test suite.

Table 2. Generated test suite for 3-way combinatorial interaction

Test Case No.	Test Case	Iteration/Child Thread No.	Max Weight	Covered pairs
T1	A1,B1,C1,D1	1/1	4	[A1,B1,C1][A1,B1,D1][A1,C1,D1][B1,C1,D1]
T4	A1,B1,C2,D2	1/1	4	[A1,B1,C2][A1,B1,D2][A1,C2,D2][B1,C2,D2]
T8	A2,B1,C1,D2	½	4	[A2,B1,C1][A2,B1,D2][A2,C1,D2][B1,C1,D2]
T9	A2,B1,C2,D1	½	4	[A2,B1,C2][A2,B1,D1][A2,C2,D1][B1,C2,D1]
T5	A1,B1,C3,D1	2/1	3	[A1,B1,C3][A1,C3,D1][B1,C3,D1]
T12	A2,B1,C3,D2	2/1	3	[A2,B1,C3][A2,C3,D2][B1,C3,D2]
T2	A1,B1,C1,D2	3/1	1	[A1,C1,D2]
T3	A1,B1,C2,D1	3/1	1	[A1,C2,D1]
T6	A1,B1,C3,D2	3/1	1	[A1,C3,D2]
T7	A2,B1,C1,D1	3/2	1	[A2,C1,D1]
T10	A2,B1,C2,D2	3/2	1	[A2,C2,D2]
T11	A2,B1,C3,D1	3/2	1	[A2,C3,D1]

The tree example shown in Fig. 2 explains how the test cases are constructed. In reality we may need only the leaf nodes and all the intermediate nodes are not used this increase the efficiency by minimising the number of nodes and giving importance only to the leaf nodes at every stage.

4 Empirical Results

With an example shown in Figure 1 at Section 4, the generated test suit (Table 2) has covered all the 3-way combinations (28) in Table 1, thus proving the correctness of the proposed strategy.

To evaluate the efficiency of the strategy for T -way test data generation, we consider six different configurations. The first three configurations have non-uniform

parametric values. The other three configurations have a uniform number of values for all parameters. The six system configurations used are summarized as follows:

- S1: 3 parameters with 3, 2 and 3 values respectively.
- S2: 4 parameters with 2,1,2 and 1 values respectively
- S3: 5 parameters with 3, 2, 1, 2 and 2 values respectively.
- S4: 3 3-valued parameters
- S5: 4 3-valued parameters.
- S6: 5 2-valued parameters.

Table 3. 2-way results

System	Exhaustive number of test cases	2-way Test suite size	2-way Reduction %
S1	18	9	50%
S2	4	4	0%
S3	24	7	70.83%
S4	27	9	66.67%
S5	81	9	88.89%
S6	32	7	78.13%

Table 4. 3-way results

System	Exhaustive number of test cases	3-way Test suite size	3-way Reduction %
S2	4	4	0%
S3	24	16	33.33%
S5	81	31	61.73%
S6	32	12	62.5%

In Tables 3 and 4, column 2 shows the exhaustive number of test cases for each system. The last column shows the percentage of reduction achieved by using our strategy.

Results in Tables 3 and 4 demonstrate that our strategy is an efficient strategy in test size reduction. In Table 3 with pairwise test suite size reduction, in some cases a high reduction is achieved, as in systems S5 and S6 (more than 75%). In case of

system S2, there is no reduction achieved because this is the minimum test suite size. In Table 4, which shows the 3-way test suite results there is reduction achieved in case of systems S3, S5 and S6, but in case of S2 no reduction is achieved as this is the minimum test suite size. The other systems such as S1 and S4 have 3 parameters only and therefore cannot be considered for 3-way test suite reductions. Thus, the tables 3 and 4 reveal that the proposed strategy works well for T-way test suite size reduction, for both parameters with uniform as well as non-uniform values.

5 Conclusion

In this paper a tree test generation strategy has been designed to support a parallel higher strength test interactions. The correctness of the proposed strategy has been proved in section 4 (Tables 2). Empirical results in Section 5 shows that our strategy is an efficient strategy in test size reduction and can generate highly reduced test suites. Our strategy includes only the minimum number of test cases which have covered the maximum number of T-way combinations into the generated test suite in each iteration, thus making it different from other strategies. Tables 3 and 4 reveal that the proposed strategy works well for different test strength (T) values, and can produce an efficient and reduced test suite size, for both uniform as well as non-uniform parametric values. Even though a good result in reduction achieved; there is a drawback in the first step of this strategy (i.e. generating the tree of test cases before reduction starts) which produce a huge number of test cases especially when the software under test has a large number of parameters. Improving will be in a future work.

References

1. Kaner, C.: Exploratory Testing. In: Proc. of the Quality Assurance Institute Worldwide Annual Software Testing Conference, Orlando, FL (2006)
2. Bryce, R., Colbourn, C.J., Cohen, M.B.: A Framework of Greedy Methods for Constructing Interaction Tests. In: Proc. of the 27th International Conference on Software Engineering, pp. 146–155. St. Louis, MO, USA (2005)
3. Tsui, F.F., Karam, O.: Essentials of Software Engineering. Jones and Bartlett Publishers, Massachusetts, USA (2007)
4. Zamli, K.Z., Klaib, M.F.J., Younis, M.I., Isa, N.A.M., Abdullah, R.: Design and Implementation of a T-Way Test Data Generation Strategy with Automated Execution Tool Support. Information Sciences Journal (2011)
5. Hartman, A., Klinger, T., Raskin, L.: IBM Intelligent Test Configuration Handler. IBM Haifa and Watson Research Laboratories (2005b)
6. Hartman, A., Raskin, L.: Combinatorial Test Services (2004a), <http://www.alphaworks.ibm.com/tech/cts> (Accessed on August 2008)
7. Zamli, K.Z., Klaib, M.F.J., Younis, M.I.: G2Way: A Pairwise Test Data Generation Strategy with Automated Execution. Journal of Information and Communication Technology 9 (2010)
8. Klaib, M.F.J., Zamli, K.Z., Isa, N.A.M., Younis, M.I., Abdullah, R.: G2Way – A Backtracking Strategy for Pairwise Test Data Generation. In: Proc. of the 15th IEEE Asia-Pacific Software Engineering Conf, Beijing, China, pp. 463–470 (2008)

9. Lei, Y., Kacker, R., Kuhn, D.R., Okun, V., Lawrence, J.: IPOG: A General Strategy for T-Way Software Testing. In: Proc. of the 14th Annual IEEE Intl. Conf. and Workshops on the Engineering of Computer-Based Systems, Tucson, AZ, U.S.A., pp. 549–556 (2007)
10. Cohen, D.M., Dalal, S.R., Fredman, M.L., Patton, G.C.: The AETG System: An Approach to Testing Based on Combinatorial Design. *IEEE Transactions on Software Engineering* 23, 437–444 (1997)
11. Cohen, M.B., Snyder, J., Rothermel, G.: Testing Across Configurations: Implications for Combinatorial Testing. In: Proc. of the 2nd Workshop on Advances in Model Based Software Testing, Raleigh, North Carolina, USA, pp. 1–9 (2006)
12. Colbourn, C.J., Cohen, M.B., Turban, R.C.: A Deterministic Density Algorithm for Pairwise Interaction Coverage. In: Proc. of the IASTED Intl. Conference on Software Engineering, Innsbruck, Austria, pp. 345–352 (2004)
13. Tai, K.C., Lei, Y.: A Test Generation Strategy for Pairwise Testing. *IEEE Transactions on Software Engineering* 28, 109–111 (2002)
14. Shiba, T., Tsuchiya, T., Kikuno, T.: Using Artificial Life Techniques to Generate Test Cases for Combinatorial Testing. In: Proc. of the 28th Annual Intl. Computer Software and Applications Conf (COMPSAC 2004), Hong Kong, pp. 72–77 (2004)
15. Cohen, D.M., Dalal, S.R., Kajla, A., Patton, G.C.: The Automatic Efficient Test Generator (AETG) System. In: Proc. of the 5th International Symposium on Software Reliability Engineering, Monterey, CA, USA, pp. 303–309 (1994)
16. Lei, Y., Tai, K.C.: In-Parameter-Order: A Test Generation Strategy for Pairwise Testing. In: Proc. of the 3rd IEEE Intl. High-Assurance Systems Engineering Symp, Washington, DC, USA, pp. 254–261 (1998)
17. Lei, Y., Kacker, R., Kuhn, D.R., Okun, V., Lawrence, J.: IPOG/IPOD: Efficient Test Generation for Multi-Way Software Testing. *Journal of Software Testing, Verification, and Reliability* 18, 125–148 (2009)
18. TConfig, <http://www.site.uottawa.ca/~awilliam/>
19. Jenny, <http://www.burtleburtle.net/bob/math/>
20. TVG, <http://sourceforge.net/projects/tvg>
21. Cohen, M.B.: Designing Test Suites for Software Interaction Testing. In: Computer Science, PhD University of Auckland New Zealand (2004)
22. Grindal, M., Offutt, J., Andler, S.F.: Combination Testing Strategies: a Survey. *Software Testing Verification and Reliability* 15, 167–200 (2005)
23. Younis, M.I., Zamli, K.Z., Mat Isa, N.A.: IRPS – An Efficient Test Data Generation Strategy for Pairwise Testing. In: Lovrek, I., Howlett, R.J., Jain, L.C. (eds.) KES 2008, Part I. LNCS (LNAI), vol. 5177, pp. 493–500. Springer, Heidelberg (2008)
24. Grindal, M.: Handling Combinatorial Explosion in Software Testing. Linköping Studies in Science and Technology, Dissertation No. 1073, Sweden (2007)

PS2Way: An Efficient Pairwise Search Approach for Test Data Generation

Sabira Khatun^{1,*}, Khandakar Fazley Rabbi¹, Che Yahaya Yaakub¹,
M.F.J. Klaib², and Mohammad Masroor Ahmed¹

¹ Faculty of Computer Systems & Software Engineering,
University Malaysia Pahang, Pahang, Malaysia

² Faculty of Science and Information Technology, Jadara University, Jordan
sabira@ump.edu.my, falzey.rabbi@ymail.com, yahaya@ump.edu.my,
masroor@ump.edu.my

Abstract. Testing is a very important task to build error free software. Usually, the resources and time to market a software product is limited, hence, it is impossible to perform exhaustive test i.e., to test all combinations of input data. Pairwise (2way) test data generation approach supports higher reduction of exhaustive numbers as well as low cost and effective. In pairwise approach, most of the software faults are caused by unusual combination of input data. Hence, optimization in terms of number of generated test-cases and execution time is in demand. This paper proposes an enhanced pairwise search approach (PS2Way) of input values for optimum test data generation. This approach searches the most coverable pairs by pairing parameters and adopts one-test-at-a-time strategy to construct final test suites. PS2Way is effective in terms of number of generated test cases and execution time compared to other existing strategies.

Keywords: Combinatorial interaction testing, Software testing, Pairwise testing, Test case generation.

1 Introduction

Software testing and debugging is one of the integral part of software development life cycle in software engineering but this process is very labor-intensive and expensive[1]. Around 50% of project money goes under software testing. Hence, focus is to find automatic and cost-effective software testing and debugging techniques to ensure high quality of released software product [2]. Nowadays research on software testing focuses on test coverage criterion design, test-case generation problem, test oracle problem, regression testing problem and fault localization problem [1]. Among these problems test-case generation problem is an important issue to produce error free software [1]. To solve this problem, Pairwise strategy (i.e. two-way interaction) has been known as an effective test case reduction strategy (and able to detect from 60 to 80 percent of the faults) [4].

* Corresponding author.

For example, for the ‘**proofing**’ tab under ‘**option**’ dialog box in Microsoft excels (Figure 1), there are 6 possible configurations needed to be tested. Each configuration takes two values (checked or unchecked), on top of that the ‘**French modes**’ takes 3 possible values and ‘**Dictionary language**’ takes 54 possible values. So to test this **proofing** tab exhaustively, the number of test cases need to be executed is $2^6 \times 54 \times 3$ i.e. 10,368. Assuming each test case may consume 4 minutes in average to execute; results around 28 days to complete the exhaustive test for this tab [3, 4].

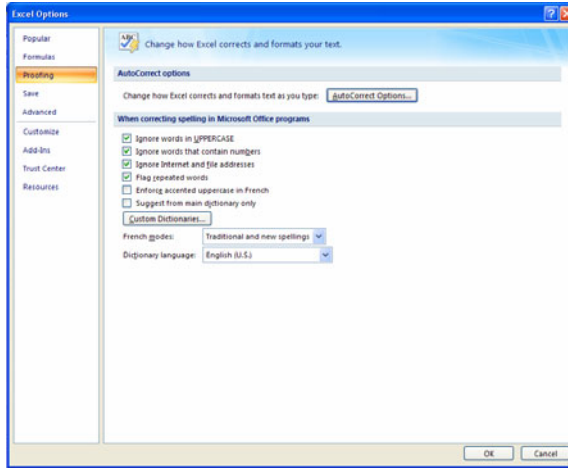


Fig. 1. Microsoft Excel Proofing Option

This is similar for hardware products as well. If a product has 30 on/off switches, to test all possible combination it may need $2^{30} = 1,073,741,824$ test cases, and consume 10,214 years by considering 5 minutes for each single test case [4]. Nowadays, research work in combinatorial testing aims to generate least possible test cases [5]. The solution of this problem is non-deterministic polynomial-time hard (NP-hard) [6]. So far many approaches have been proposed and also many tools have been developed to find out the least possible test suit in polynomial time [5-8, 10-14] but yet optimum one is in need. This paper introduces an enhanced pairwise search approach (PS2Way) to generate pairwise test data in terms of optimum size and time consumption.

The paper is organized as follows. The detail related work is described in Section 1.1. Followed by, the proposed PS2Way algorithm details, the empirical results with discussions and comparison, and finally the conclusion.

1.1 Related Work

Empirical facts show that lack of testing for both functional and nonfunctional is one of the major sources of software and systems bug/errors [7, 8]. National Institute of Standard and Technology (NIST) estimated that the cost of software failure to the US

economy at $\$6 \times 10^{10}$, which was the 0.6 percent of GDP [9, 10]. It was found that more than one-third of this cost can be reduce by improving software testing structure. Hence, automatic testing is of critical concern [11]. In the course of automation, software can become more practical and scalable. However, the automated generation of test case is challenging [12]. The underlying problem is known to be un-decidable and NP-hard thus researchers have focused on the techniques that search to find near optimal test sets in a reasonable time [13, 14].

Pairwise testing becomes a significant approach to software testing as it often provides efficient error detection at a very low cost. It keeps a good balance between the magnitude and effectiveness of combinations. It requires every combination of any two parameter values to be covered by at least one test case [14, 15]. From the point of pairwise there are some pre-defined rules to calculate the test cases directly from the mathematical functions, which are known as algebraic strategy [4]. On the other side, computational approaches are based on the calculation of coverage of generated pairs, followed by an iterative or random search technique to create test cases.

IRPS algorithm [6] uses the computational approach. It is a deterministic strategy which generates all pairs and then stores it to the linked list. Finally it searches the entire list, select best list and empties the list. When all list become empty, the collection of best list is determined as the final test suite.

The Automatic Efficient Test Generator (AETG) [16] and its deviation mAETG [6] generate pairwise test data using computational approach. This approach uses the 'Greedy technique' to build test cases based on covering as much as possible uncovered pairs. AETG uses a random search algorithm [17]. Genetic Algorithm (GA) and Ant Colony Algorithm (ACA) are the variants of AETG [18, 19]. Genetic algorithm [18] creates an initial population of individuals (test cases) and then the fitness of those individuals is calculated. Then it starts discarding the unfit individuals by the individual selection methods. The genetic operators such as crossover and mutation are applied on the selected individuals and this continues until a set of best individuals found. Ant Colony Algorithm [18] candidate solution is associated with the start and end points. When an ant chooses one edge among the different edges, it would choose the edge with a large amount of pheromone which gives the better result with the higher probability.

The In-Parameter-Order (IPO) [12] strategy starts with an empty test set and adds one test at a time for pairwise testing. It creates the test cases by combination of the first two parameters, then add third and calculate how many pair is been covered and so on until all the values of each parameter is checked. This approach is deterministic approach.

AllPairs [20] algorithm can generate test suites covering all pairwise interactions within a reasonable time. The strategy seems to be deterministic strategies since the same test suite are generated every run time.

The Simulated Annealing (SA) [21] algorithm is also a deterministic strategy with the same generated test suite for every run time.

Generalization of Two Way test data (G2Way) [22] is one of the excellent tools based on computational and deterministic strategy. It is based on backtracking algorithm and uses customized markup language to describe base data. The G2Way backtracking algorithm tries to combine generated pairs so that it covers highest pairs. Finally after covering all the pairs, the test case treats as a final test suite.

2 Proposed PS2Way Strategy

The proposed algorithm works as follows: First, it creates pair parameters and their values. Then values of one pair is combined with another pair by calculation the highest possible coverable pairs. In this way it constructs a test case and adds to the final test suit. To make this easily understandable, a scenario is presented in terms of example as follows:

Table 1. Example parameters with values

Parameters	A	B	C	D	E	F
Values	a1	b1	c1	d1	e1	f1
	a2	b2	c2	d2	e2	f2

In Table 1, there are 6 parameters as: A, B, C, D, E, and F, each having 2 values. The PS2Way algorithm first generates pair parameters which are AB, CD, EF and then generate the exhaustive test cases as shown in Table 2.

Table 2. Pair parameters with exhaustive test cases

Pair Parameters	AB	CD	EF
Generated all possible test cases	[a1, b1]	[c1, d1]	[e1, f1]
	[a1, b2]	[c1, d2]	[e1, f2]
	[a2, b1]	[c2, d1]	[e2, f1]
	[a2, b2]	[c2, d2]	[e2, f2]

Each AB pair tries to combine with one CD pairs. If combined pairs give highest coverage or maximum coverage, then it tries to combine with the values of EF pairs. So the test case generation approach is in greedy manner and constructs test cases one at a time.

From Table 2, PS2Way tries to combine [a1, b1] with 4 possible values of CD in the list [[c1, d1], [c1, d2], [c2, d1], [c2, d2]]. The first pair which gives the highest

coverage looks for the available pairs again which is [[e1, f1], [e1, f2], [e2, f1], [e2, f2]] as shown in Table 3. The highest coverage will then add to the final test suit.

Table 3 shows, one of AB pairs [a1, b1] searches for the best pairs among the available pairs of CD and its output should be only one, which is [a1, b1, c1, d1] as the first uncovered final test-case with full coverage. Again, generated pair [a1, b1, c1, d1] search for the available pairs of EF and the generated output is [a1, b1, c1, d1, e1, f1]. Same procedure is followed by other AB pair parameters to generate other final test cases. The highest coverable pairs are stored on the final test set. Figure 2 shows the test cases generation algorithm and Figure 3 shows the corresponding flowchart.

Table 3. Example of pair search and final test case generation

Initial pairs	Available pairs	Best uncovered pairs	Available Pairs	Best uncovered pairs
[a1, b1]	[c1, d1]	[a1, b1, c1, d1]	[e1, f1]	[a1, b1, c1, d1, e1, f1]
	[c1, d2]		[e1, f2]	
	[c2, d1]		[e2, f1]	
	[c2, d2]		[e2, f2]	
[a1, b2]	[c1, d1]	[a1, b2, c1, d2]	[e1, f1]	[a1, b2, c1, d2, e1, f2]
	[c1, d2]		[e1, f2]	
	[c2, d1]		[e2, f1]	
	[c2, d2]		[e2, f2]	
[a2, b1]	[c1, d1]	[a2, b1, c2, d1]	[e1, f1]	[a2, b1, c2, d1, e2, f1]
	[c1, d2]		[e1, f2]	
	[c2, d1]		[e2, f1]	
	[c2, d2]		[e2, f2]	
[a2, b2]	[c1, d1]	[a2, b2, c2, d2]	[e1, f1]	[a2, b2, c2, d2, e2, f2]
	[c1, d2]		[e1, f2]	
	[c2, d1]		[e2, f1]	
	[c2, d2]		[e2, f2]	

3 Empirical Results

To evaluate the efficiency of our algorithm (PS2Way), for pairwise test data generation, we have considered 5 different system configurations. Among those the first 3 are uniform parameterized values and the rest are non-uniform as follows:

S1: 3 3-valued parameters,
 S2: 4 3-valued parameters,
 S3: 13 3-valued parameters,
 S4: 10 5-valued parameters,
 S5: 1 5-valued parameters, 8 3-valued parameters and 2 2-valued parameters.

The consideration of the parameters and assumptions are according to some of the related existing algorithms that support pairwise testing to compare our results with those.

Table 4 shows the comparison of generated test suite size by our algorithm PS2Way with others. The shadowed cells indicate the best performance in term of generated test case size. It shows that proposed PS2Way produces the best results in S1, S2, and S4 (shaded) except S3 and S5. However test case production is a NP-complete problem and it is well known that no strategy may perform the best for all cases. It shows three best cases, which is highest among all related algorithms.

Algorithm to Generate Test Suits ()

```

Begin
  Let  $P_p = \{\}$  represents the set of all possible pairs
  Let  $P_s = \{\}$  represents the pairs where all the  $P_s$  stores
  in  $P_p$ 
  Let  $P_b = \{\}$  represents the best pairs set which cover
  highest pairs
  Let  $P_f = \{\}$  as empty set represents the Final test suits
  Let  $C_b$  as number = 0 represents the best covering number
  Let  $C_c$  as number = 0 represents the current covering
  number
  For each  $P_s$  as  $P_1$  in  $P_p$ 
    For each next  $P_s$  as  $P_2$  in  $P_p$ 
      Add  $P_1$  with  $P_2$  and put in  $P$ 
       $C_c =$  Get coverage pair number of  $P$ 
      IF  $C_b$  is less or equal to  $C_c$ 
        Put  $C_c$  into  $C_b$ 
        Put  $P_2$  into  $P_b$ 
      End IF
    End For
  Add  $P_1$  with  $P_b$  and store to  $P_f$ 
End For
End

```

Fig. 2. PS2Way pseudo code for test case generation

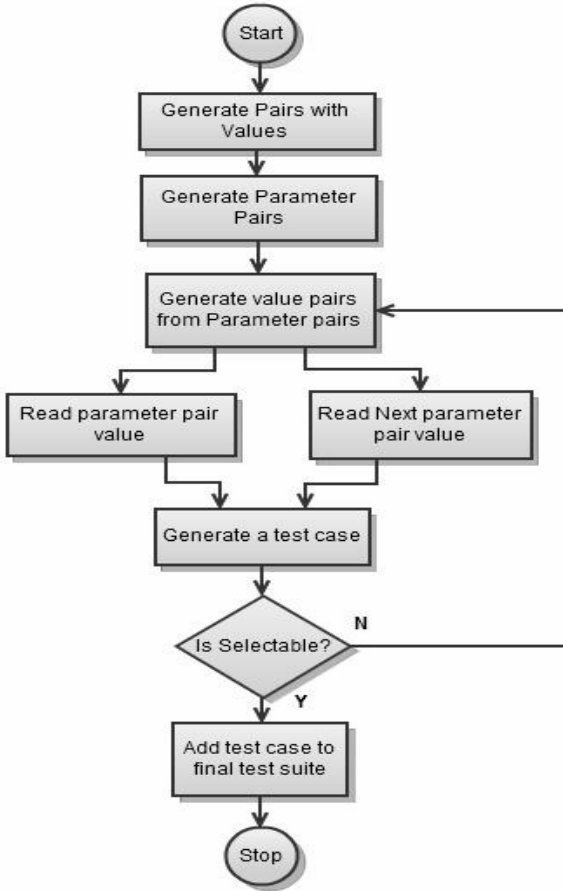


Fig. 3. Flow chart of the Algorithm

Table 4. Comparison based on the generated test size

Sys	AETG [16]	AETGm [6]	IPO [12]	SA [21]	GA [18]	ACA [19]	ALL Pairs [20]	G2Way [22]	Proposed PS2way
S1	NA	NA	NA	NA	NA	NA	10	10	10
S2	9	11	9	9	9	9	10	10	9
S3	15	17	17	16	17	17	22	19	24
S4	NA	NA	47	NA	NA	NA	49	46	46
S5	19	20	NA	15	15	16	21	23	24

For a fair comparison of execution time (i.e., complexity) among related test strategies, either computing environment should be same (usually which is not possible) or need the source code (also not available most of the cases). All Pairs tool [20] is free to download and can execute using any platform. Hence we have managed to compare the execution time of R2Way with it using the same platform as follows: Intel P IV 3 GHz, 1 GB RAM, Java programming language, and Windows XP as OS.

The results in Table 5 show that the execution time for PS2Way is acceptable and better than AllPairs [20] most of the cases.

Table 5. Comparison Based on Execution Time (in seconds)

<i>Sys</i>	<i>ALL Pairs [20]</i>	<i>PS2Way</i>
S1	0.08	0.027
S2	0.23	0.08
S3	0.45	0.2
S4	1.05	1.02
S5	0.35	0.58

4 Conclusion

In this paper we have proposed pair parameter based search algorithm (PS2Way) for test case generation for pairwise testing. PS2Way combines two parameters together (a pair) and search for another pair. The architecture and the algorithm is a far different than other existing algorithms because the parameters create the pair among themselves first and all the pairs look for other pairs to obtain the highest coverage. The strategy of this algorithm is to generate test cases from the parameter pairs. The correctness of the proposed strategy is apparent. The algorithm is efficient in terms of execution time and able to generate highly reduced test suites to fulfill the current demand by software development companies. The proposed algorithms could be further extended to support higher t-way interaction testing which is under investigation in University Malaysia Pahang.

References

1. Chen, X., Gu, Q., Qi, J., Chen, D.: Applying Particle Swarm optimization to Pairwise Testing. In: proceedings of the 34th Annual IEEE Computer Software And Application Conference, Seoul, Korea (2010)
2. Cui, Y., Li, L., Yao, S.: A New strategy for pairwise test case generation. In: Proceedings of The Third International Symposium on Intelligent Information Technology Application, NanChang, China (2009)

3. Lei, Y., Kacker, R., Kuhn, D.R., Okun, V., Lawrence, J.: IPOG: A general strategy for t-way software testing. In: proceedings of the 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, Tucson, Arizona (2007)
4. Younis, M.I., Zamli, K.Z., Mat Isa, N.A.: Algebraic Strategy to Generate Pairwise Test Set for Prime Number Parameters and Variables. In: proceedings of the IEEE international conference on computer and information technology, Kuala Lumpur, Malaysia (2008)
5. Klaib, M.F.J., Muthuraman, S., Ahmad, N., Sidek, R.: A Tree Based Strategy for Test Data Generation and Cost Calculation for Uniform and Non-Uniform Parametric Values. In: Proceedings of the 10th IEEE International Conference on Computer and Information Technology, West Yorkshire, UK (2010)
6. Younis, M.I., Zamli, K.Z., Mat Isa, N.A.: IRPS - An Efficient Test Data Generation Strategy for Pairwise Testing. In: Proceedings of the 12th international conference on Knowledge-Based Intelligent Information and Engineering Systems. LANI, Springer, Heidelberg (2008)
7. Leffingwell, D., Widrig, D.: Managing Software Requirements: A Use Case Approach. Addison-Wesley, Reading (2003)
8. Glass, R.L.: Facts and Fallacies of Software Engineering. Addison-Wesley, London (2002)
9. National Institute of Standards and Technology, The Economic Impacts of Inadequate Infrastructure for Software Testing, Planning Report, May 2-3 (2002)
10. Harman, M., McMinn, P.: A Theoretical and Empirical Study of Search-Based Testing: Local, Global, and Hybrid Search. IEEE Transactions on Software Engineering 36(2), 226–247 (2010)
11. McMinn, P.: Search-Based Software Test Data Generation: A Survey. Software Testing, Verification and Reliability 14(2), 105–156 (2004)
12. Lei, Y., Tai, K.C.: In-Parameter-Order: A Test Generation Strategy for Pairwise Testing. In: Proceedings of the 3rd IEEE International conference on High-Assurance Systems Engineering, Washington, DC, USA (1998)
13. Gong, D., Yao, X.: Automatic detection of infeasible paths in software testing. IET Software 4(5), 361–370 (2010)
14. Kim, J., Choi, K., Hoffman, D.M., Jung, G.: White Box Pairwise Test Case Generation. In: Proceedings of the IEEE Seventh International Conference on Quality Software, Oregon, USA (2007)
15. Soh, Z.H.C., Abdullah, S.A.C., Zamli, K.Z.: A Parallelization Strategies of Test Suites Generation for t-way Combinatorial Interaction Testing. In: Proceedings of the IEEE International Conference on Information Technology, International Symposium, Kuala Lumpur, Malaysia (2008)
16. Cohen, D.M., Dalal, S.R., Fredman, M.L., Patton, G.C.: The AETG System: An Approach to Testing Based on Combinatorial Design. IEEE Transactions on Software Engineering 23(7), 437–444 (1997)
17. Harman, M., Jones, B.F.: Search-based Software Engineering & Information and Software Technology pp. 833-839 (2001)
18. Shiba, T., Tsuchiya, T., Kikuno, T.: Using Artificial Life Techniques to Generate Test Cases for Combinatorial Testing. In: Proceedings of the 28th Annual Int. Computer Software and Applications Conf (COMPSAC 2004), Hong Kong (2004)
19. Chen, X., Gu, Q., Zhang, X., Chen, D.: Building Prioritized Pairwise Interaction Test Suites with Ant Colony Optimization. In: Proceedings of the 9th International IEEE Conference on Quality Software, Jeju, Korea (2009)

20. Bach, J.: Allpairs Test Case Generation Tool,
<http://tejasconsulting.com/open-testware/feature/allpairs.html> (last access September 27, 2009)
21. McCaffrey, J.D.: Generation of Pairwise Test Sets using a Simulated Bee Colony Algorithm. In: Proceedings of The IEEE International Conference on, Information Reuse & Integration, Las Vegas, USA (2009)
22. Klaib, M.F.J., Zamli, K.Z., Isa, N.A.M., Younis, M.I., Abdullah, R.: G2Way – A Backtracking Strategy for Pairwise Test Data Generation. In: Proceedings of The 15th IEEE Asia-Pacific Software Engineering Conf., Beijing, China (2008)

The Preferable Test Documentation Using IEEE 829

Roslina Mohd Sidek¹, A. Noraziah¹, and Mohd Helmy Abd Wahab²

¹ Faculty of Computer Systems and Software Engineering
Univerisiti Malaysia Pahang
Lebuhraya Tun Razak, 26300 Kuantan, Pahang, Malaysia
{roslinams, noraziah}@ump.edu.my

² Universiti Tun Hussein Onn Malaysia
86400 Pt. Raja, Batu Pahat, Johor, Malaysia
helmy@uthm.edu.my

Abstract. During software development, testing is one of the processes to find errors and aimed at evaluating a program meets its required results. In testing phase there are several testing activity involve user acceptance test, test procedure and others. If there is no documentation involve in testing the phase the difficulty happen during test with no solution. It because no reference they can refer to overcome the same problem. IEEE 829 is one of the standard to conformance the address requirements. In this standard has several documentation provided during testing including during preparing test, running the test and completion test. In this paper we used this standard as guideline to analyze which documentation our companies prefer the most. From our analytical study, most company in Malaysia they prepare document for Test Plan and Test Summary.

Keywords: Documentation, Testing, Bug, Test Plan, Software Test Documentation.

1 Introduction

Defects is become a major problem for all developer during software development. The defect should be solved as many as it can to make end product reliable, usable and others. To detect the defect is in testing phase which called software testing. Software testing is very important to make software free error. In testing stages the tester should document all testing activities to improve the testing activity easier for next time. If there is lack of documentation a lot of problem will occur. As mentioned by Roslina Mohd Sidek et al [1] regarding all difficulties if there is no well prepare documentation for testing example product delay, wasting time and money. If documentation is not applied during the testing, it makes the finding bug become more complicated. R. Kumar [2] said that careful documentation can save an organization's time and money. It is because of finding the bug will make cost of correction becomes higher. According to Software QA Associates[2][3] in the internet era, in organizations with little software configuration management, projects

are often built and tested without adequate documentation for example, no formal test planning, no test procedures, etc. Refer to L.P David et al said [4] to conduct inspection and test effectively with good documentation. The precise documents will allow engineers to authorize all component meets its stated specification, and the product will be more satisfactory. In K.Hertel et al [5] use photo in documentation purposely preparation for new colleagues who are not familiar with the test project or with the daily work routines.

Software testing states a number of rules that can serve as well as testing objectives: Testing is a process of executing a program with the intent of finding an error; good test case is one that has a high probability of finding as many as we can; successful test is one that uncovers an as-yet-undiscovered error. The aim of software testing according Sommerville [6] is discover defects by testing individual program components. The components may be functions, objects or reusable components. the paper discuss about IEEE 829 standard in Section 2. The factors of software test documentation are discussed in Section 3 and discussion in Section 4 and conclusion in Section 5.

2 Literature Review

In this section we discuss the software test documentation according to IEEE 829 Documentation [7]. The IEEE 829 is a standard that specifies the form of a set of documents for use in eight defined stages of software testing, each stage possibly making its own separate type of document. The standard specifies the format of these documents but does not require whether they all must be produced, nor does it include any criteria regarding adequate content for these documents. These are a matter of decisions outside the purview of the standard. This standard is one of the standards of conformance to address the requirements to which a software developer must conform. Other standard such as IEEE 828 used for software configuration, IEEE 1012 used for verification and validation and others. The IEEE 829 has several types of document for testing which can be used in three distinct phases of software testing. The types of document are preparation of tests, running the test and completion of testing.

2.1 IEEE 829[8]

In this IEEE 829 divided into 3 sections which are preparation of tests, running the test and completion the test.

2.1.1 Preparation of Tests

The most important part of any software testing is preparation of testing. The purpose of this state is to prepare an effective and efficient set of test, and create the environment for them to run in. In preparation test the test involves are Test Plan, Test Design Specification, Test Case Specification, Test Procedure and Test Item Transmittal Report. The functions of those tests are in the Table 1.

Table 1. The functions of Test Plan in preparing test

Type	Function
Test Plan	Plan how the testing will proceed.
Test Design Specification	Decide what needs to be tested. Describe the test to be performed on a specific feature.
Test Case Specification	Create the test to be run. It also explains the value or condition will be sent to software and result expectation.
Test Procedure	Describe how the tests are run or how it performs
Test Item Transmittal Report	Specify the items released for testing

The Test Plan is the crucial document around which all the software testing projects revolve. It describes the activity should be done, what quality standard need to achieve, resource that we need to support quality, time scale, and the risks management and the solution plan. The creating of Test Design is the step to develop test project. It records all test details and derived from documents requirement and designs phase. The test case specification is produced when the test design is completed. It specifies for testing requirement such as exact input and output values of any standing data also all steps to set up the tests. If we not define all this expected value it may results very poor quality set of test cases. The test procedures are developed from both Test Design and Test Case Specification. The test procedure describes how the test officer will run the test, physical set-up required and the procedure step the tester need to be followed. Test Item Transmittal describes the item being delivered for testing, where to find the item, new about item and approval for the release. The crucial thing of the document is to provide the test officer a warranty that the item are fit to be tested and give a clear mandate to start testing.

2.1.2 Running the Tests

Starting from this phase is the documentation for running the test. When the tests have been developed, then they can be run. The test result should be recorded in the Test Log and Test Incident Report.

Table 2. Test while running the tests

Test	Functions
Test Log	Record the details of tests in time order.
Test Incident Report	Record details of events that need to be investigated.

The Test Log is function to record the details of test cases in time order. The details are the Test Cases have been run, the order of running activity and test results. The results are either the test passed or fail. If the result is passed then document it

whether get actual result and expected result and if fail is there a discrepancy. If a discrepancy more than one then the Test Incident Report raised or updated and identities recorded on the Test Log. It allows progress of the testing to be checked and the cause will be found the incident information out. The Test Incident Report is also named incident report. It because a discrepancy between expected and actual results can occurs for a number of reasons included expected results being wrong, test being wrong or inconsistency in the requirements meaning.

2.1.3 Completion of Testing

Eventually, testing will be completed according to the criteria specified in the Test Plan. This is when the success or failure of the system is decided based on the results.

Table 3. Test when completion the test

Type	Function
Test Summary Report	Summarize and evaluate tests.

The Test Summary brings together all relevant information about the testing, including an assessment about how well the testing has been done, the number of incidents raised and outstanding, and critically an assessment about the quality of the system. It also recorded for use in future project planning is details of what was done, and how long it took. This document is important in deciding whether the quality of the system is good enough to allow it to proceed to another stage.

2.2 Related Works

Rodziah[9] also was doing survey in test documentation in year 2001. She did Software Test Documentation in her master thesis. She mentioned to cover all industries sector, University, Research Institute and Public Administration. The problem in her thesis is there is no result in Research Institute and Public Administration. The area that she covered was too broad but the respondent that she get not cover all sector that mentioned.

K.Hertel et al [5] divides to follow the structure described above and is divided into several parts incoming inspection such as on transport frame, assembly into test support. Tests after installation such as open cryostat, coil mounted in support and everything inside the cryostat, tests in closed cryostat such as ambient temperature, vacuum, cool down and others. Fig 1 shows of the amendment on the basis of these instructions a certain test-structure was created, which considers the instructions' requirements as well as the technical abilities of the test facility at Commissariat a l'Energie Atomique (CEA) in Saclay, France and the work cycle for the necessary assemblies and installations.

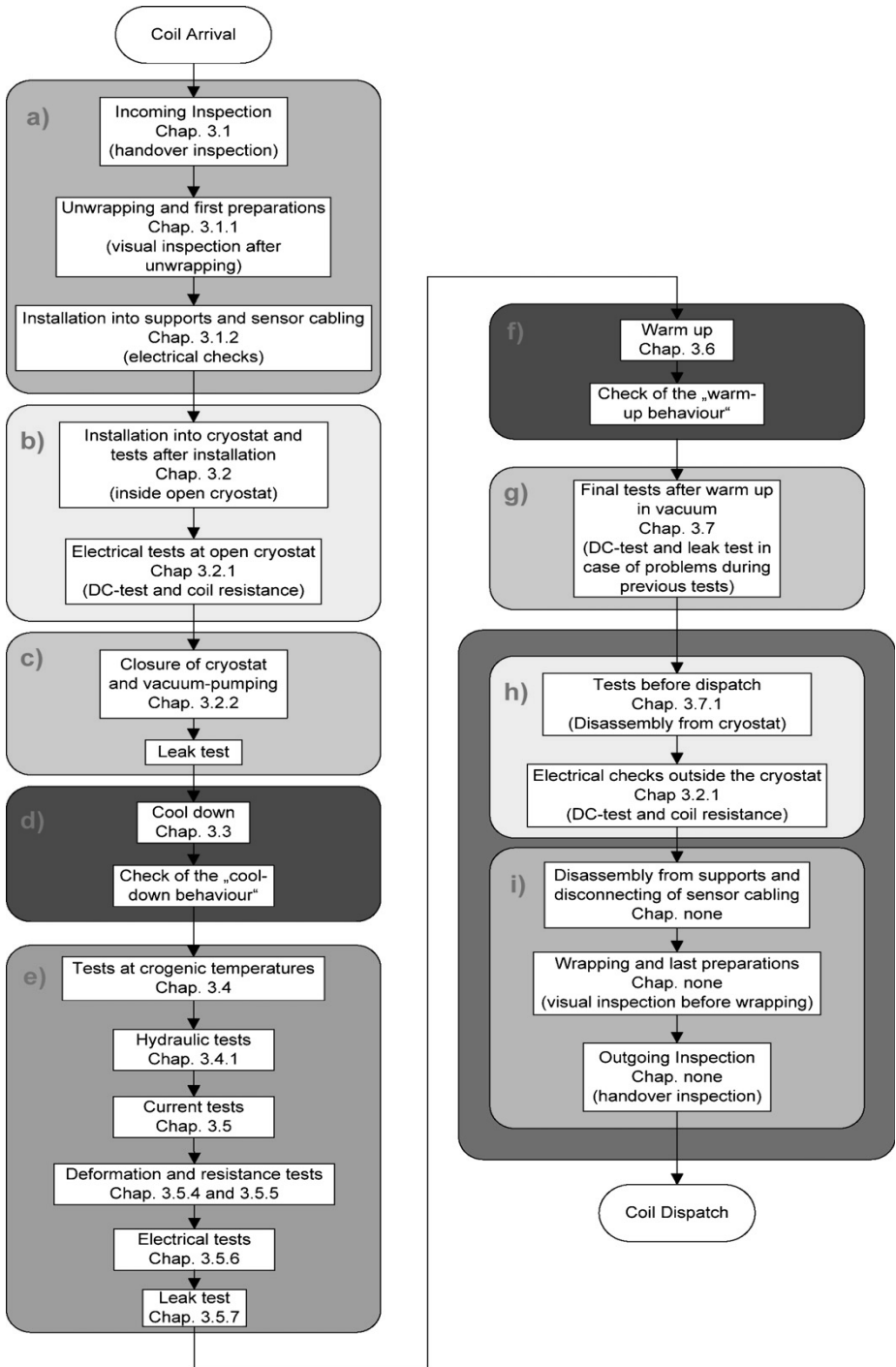


Fig. 1. The amendment of test

3 Data Benchmark

Our target companies are the company which involved with the software development. We get the result from several companies as our target respondent. We assume if the percentage of data collective is like in Table 6.

Table 4. Survey Benchmark

Percentage	Description
51-100	Practice
0 - 50	Not Practice

Refer to our study most of our target companies are practice certain part in the software test documentation. During the software testing there are so many problems that have to be faced. This paper concentrates only on the test plan. From the analysis, we identify that most companies have problems during testing because more products being developed nowadays, faster application development, get instruction just convey by word of mouth, recurring problems occurred, and fast growing organization.

4 Analysis of Software Test Documentation

We have been discussed all elements that proposed by IEEE 829 Documentation. We distribute questionnaires to several companies in Malaysia to acquire the data. The type of response should in nominal [10]. So, the respondent option answer the questionnaire is Yes, No or Not Sure. From the analytical study of software test documentation in selected companies in Malaysia, majority of the companies have done the documentation. In our questionnaires we focus on the test documents that have been produced:

Table 5. The Variable for the Test Documentation

Variable	Descriptions	Type
Q1.5	Are you written test documents available?	nominal
Q1.6	If YES – Which one of the following test documents do you produced?	
Q1.6.1	Test Plan	nominal
Q1.6.2	Test Design Specifications	nominal
Q1.6.3	Test Case Specifications	nominal
Q1.6.4	Test Procedures	nominal
Q1.6.5	Test Item Transmittal Report	nominal
Q1.6.6	Test Log	nominal
Q1.6.7	Test Incident Report	nominal
Q1.6.8	Test Summary Report	nominal

With reference to our analytical study, we get the result shown in Table 5.

Table 6. Result of in Percentage

No	Test Document	Percentage
1	Test Plan	73.7
2	Test Design Specifications	47.4
3	Test Case Specifications	47.4
4	Test Procedures	52.6
5	Test Item Transmittal Report	47.4
6	Test Log	52.6
7	Test Incident Report	26.3
8	Test Summary Report	57.9

Software Test Documentation concentrates on the documentation or written test including Test Plan. From the analytical studies we found the companies prefer to produce test documents of Software Test Documentation are Test Plan with 73.7%, Test Procedure with 52.6%, Test Log with 52.6%, and Test Summary Report with 57.9%. The result is shown in Fig 4.

Test Plan is one of the test documentations in software testing. It is useful on a daily or hourly by the testers performing the testing. The Test Plan should be well plan to make it success in implementing the test.

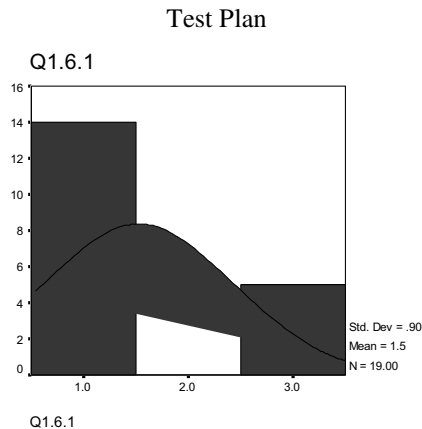


Fig. 4. Frequency for Test Plan

The frequency of Test Procedure is shown in Fig 5. It is also one of the test documents in software test documentation. Most of the selected companies in Malaysia identified the steps required to operate the system and exercise the specified test cases in order to implement test design.

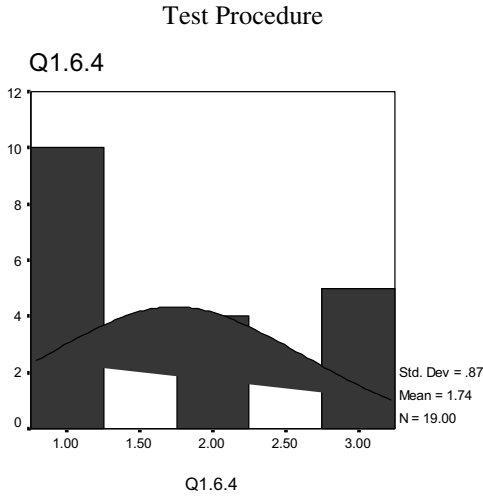


Fig 5. Frequency for Test Procedure

The Test Log frequency is shown in Fig 6. With this Test Log all activities is recorded in time order. Most of the target companies have recorded the testing activitie

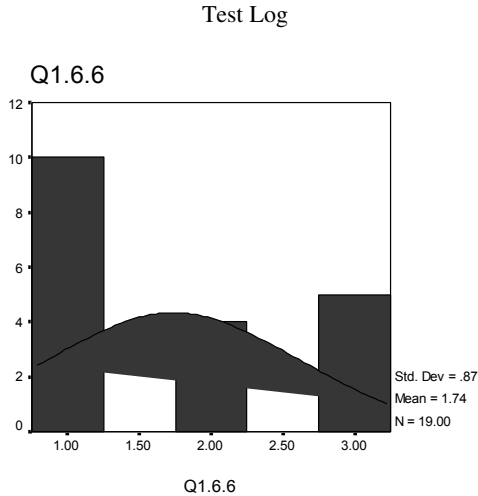


Fig. 6. Frequency for Test Log

The Test Summary Report frequency is shown in Fig 7. Most of the targets companies are implement the summary report because the software has to be delivered to customer. So, it should be tested. The tester finalized the test to make sure the bug that affect the software have been fixed and secured.

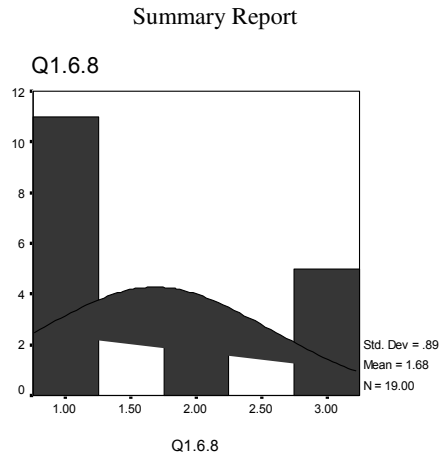


Fig. 7. Frequency for Test Summary Report

5 Conclusion

This paper presented the test documents of software test document in selected company in Malaysia. The test documents are refers to the IEEE 829 Documentation. In our analytical study described has confirmed the existence of the use of Software Test Documentation in testing certain activities. The documentation the most implements are Test Plan, Test Log, Test Procedure and Test Summary Reports. Test Plan is an idea of activities during testing. Some organizations have their own documentation standard according to their organization policy. The standards that mostly referred are CMMI and ISO. The test documentation implemented almost all from the company that respond to this analytical study. Implementing standard certification gives companies in Malaysia toward achieving higher quality in software. It really gives impact to software industry in Malaysia in fast delivery no delay because developer can refer to the documentation which is has same bug problem and so forth. At the same time, our software industry in Malaysia can maintain the cost estimation for training to send new employee which just joined the project. They just can refer to the test documentation how to solve it.

References

1. Sidek, R.M., Ahmad, N.: Software Test Documentation. In: Proceeding of Software Engineering and Computer Systems (ICSECS 2009), Kuantan, pp. 599–604 (2009)
2. Kumar, R.: Documentation and QA (2002), <http://www.stickminds.com> (viewed March 2005)
3. Software QA Associates, QA/Test Plans and Procedures: The Value of Software Test Documentation (2001), <http://sqa-associates.com> (viewed March 2005)
4. David, L.P., Sergie*, A.V.: Precise Documentation of Critical Software In: 10th IEEE High Assurance Systems Engineering Symposium, pp. 237–344. USA (2007)

5. Hertel, K.: Documentation and first data-analysis on acceptance tests of W7-X coils. *Fusion Engineering and Design* 84, 943–948 (2009),
<http://www.elsevier.com/locate/fusengdes>
6. Sommerville.: *Software Engineering* 8. Pearson Education Limited, England (2007)
7. Galin, D.: *Software Quality Assurance*. p. 509. Pearson Addison Wesley (2004); Ron, P.: *Software Testing*. p. 13. SAMS publishing (2004)
8. *Information and advice for software testing* (2011),
<http://www.coleyconsulting.co.uk/IEEE829.htm>
9. Joachim, B., Dirk, M.: A View-Based Approach for Improving Software Documentation Practices. In: *Proceedings of the 13th Annual IEEE International Symposium and Workshop on Engineering of Computer Based Systems (ECBS 2006)*, Germany (2006)
10. Wikipedia (2011),
http://en.wikipedia.org/wiki/Level_of_measurement#Nominal_scale (accessed April 2011)

Automatic Analysis of Static Execution Time for Complex Loop Contained External Input

Yun-Kwan Kim¹, Doo-Hyun Kim^{2,*}, Tae-Wan Kim³, and Chun-Hyon Chang¹

¹ Dept. of Computer Science, Konkuk University

² School of Internet and Multimedia Engineering, Konkuk University,
1 Hwayang-dong Gwangjin-gu Seoul, Korea

³ Dept. of Electrical Engineering, MyongJi University,
San 38-2 Namdong, cheoin-gu, Yongin, Gyeonggido, Korea
{apost1ez, doohyun}@konkuk.ac.kr, twkim@mju.ac.kr,
chchang@konkuk.ac.kr

Abstract. Analyzing execution time in static manner is tedious, due to unbounded external inputs in loops and control flow. In the past decades there has been substantial research undergone for applying user input in various manners. One of them, Parametric WCET (Worst-Case Execution Time) analysis uses parameter to get a user input. It can give detailed formula expressed in the input variables of a program during analysis time. This can help an analyzer to offer more accurate and flexible result. However, there are problems to analyze restricted loop and, to provide simple loop bound. For this reason, it needs a measure to analyze varied structural loops and detailed loop bound applying feasible paths. In this paper we present automatic analysis of parametric static execution time for complex loop contained external input. Our proposed method, using control variable information table, can analyze complex structural loop that was difficult to analyze before and, offers accurate and flexible result better than existing manner.

Keywords: Parametric WCET Analysis, loop bound, real-time.

1 Introduction

In a real-time system, timing reliability is most important to guarantee response time without a failure of tasks and violations [1, 3]. Accordingly, to guarantee timing reliability, developers need a point of reference to determine deadline and check temporal violations in development process. To present this point, researchers need to analyze in a static manner for Worst-Case Execution Time.

But it has limited possible platform to analyze and has over estimation to analyze execution time in the static manner because it depends on hardware specification offered by vendors. Additionally, there are problems for unsettled loop bound by user input and environment variables. So it needs user input during analysis for external factors of codes [2]. It is difficult for automation of analysis, there has been some research undergone for applying user input.

* Corresponding author.

Research has been undertaken to solve these problems with transforming single-path program [11, 12, 13] to reduce effects of input and user annotation to input external data from developers for analysis [14, 15]. However, it is still hard to take an input dependency away completely, and to analyze varied execution paths caused by fixed user annotations. They should cause an incorrect analysis result [9]. Therefore, the execution time analysis needs to apply effects of external inputs in program flow and to be flexible, not fixed, in responding to input from user.

One of recent study, Parametric WCET analysis using parameter to get a user input can give detailed formula expressed in input variables of a program during analysis time. This can help an analyzer to offer more accurate and flexible result. However, there are problems to analyze restricted loops and provided just simple loop bound. Because it analyzes simple increase and decrease of integer type indexes and unfeasible inside flow in loops.

For this reason, it is necessary to measure and analyze varied structural loops and detailed loop bound applying internal feasible paths of loops. In this paper we present automatic analysis of parametric static execution time for complex loop contained external input. Our proposed automation method uses E-CFG (External Input Control Flow Graph) [7] and control variable information table [18]. It can analyze complex structural loop which is highly difficult to analyze with other methods. Our approach offers accurate and flexible result better than existing technique using abstract domain \langle interval, congruence \rangle . This can be achieved by analyzing complex structured program that may not exist in parametric WCET analysis, and reduce user input, thereby helping to get exact execution time through using detailed information.

The rest of the paper is organized as follows. Section 2 gives a summary on related research of static execution time analysis and abstract interpretation as basis of analysis. Section 3 describes our approach to analyze feasible path in loops and complex loop using control variable information table. Section 4 presents some evaluations of the approach. Finally, Section 5 concludes the paper and gives remarks about future work.

2 Related Work

2.1 Static WCET Analysis

Static execution time analysis means predicting the execution time of application from source code or executable binary without execution. The Static analysis technique has unique feature which can analyzes in various aspects of targets with less time and cost even if it has insufficient accuracy as compared with executing measurement. Generally, a goal of the execution time analysis is to get Worst-Case Execution Time (WCET). A WCET must need to guarantee stability of hard real-time system that considers particularly, timely and predictability as being most important, and to affect decision of schedulability.

There are several analysis tools developed globally to achieve such goal, such as aiT WCET analyzer, Bound-T, HEPTANE [2, 7, 10]. Figure 1 illustrates a common process of static execution time analysis.

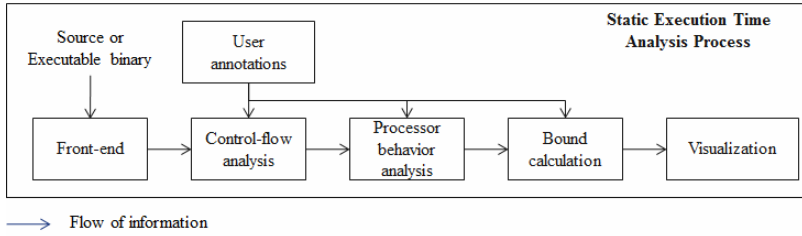


Fig. 1. General process of Static Analysis

Static execution time analysis is composed of Front-end with dependent on source code, Control-Flow Analysis, Bound calculation and Processor behavior analysis with dependent on target platform. In analysis process, user annotation provides non-existent information in source code though it is need to execution.

In a series of analysis process, it needs additional input and annotations by developers due to non-existent data in source. This information includes the user input values, specification of following execution paths, and loop bound of iteration. Consequently, researches are working with annotation language to provide the information systematically [14, 15] and single path programming to reduce the effect from external inputs [13]. However, the complicity for analysis still exists. In the above works, researches have used input by user or tools before analysis started to solve nonexistent data problem. Those input data is fixed before analyzing which makes difficult for developers to identify the changes in execution time according to varied data input.

At the same time, parametric WCET analysis attains input using parameter for necessary information in analysis process. The analysis is general, fully automatic and works for arbitrary control flow and can give potentially very complex and detailed formulae expressed in the input variables of a program [16].

A method using parametric WCET analysis is to analyze simple structural program but it is difficult to analyze complex structure or nested loops programs [16, 17]. Moreover it should contain unfeasible path through unconcern branch that is generated by external input as considering all range of input in a flow analysis. Thus, it is necessary to analyze the complex structural programs automatically. Furthermore, the hardest problem of execution time analysis is inspecting the feasible path in detail to reduce the estimation time.

2.2 Abstract Interpretation

Abstract interpretation is a technique for program analysis to analyze execution time from calculating run-time behavior of a program without running all input data which guarantees termination of calculation. It is to calculate information of the program behavior using value description or abstract values instead of real values. Abstract interpretation has three important properties:

1. It yields an approximate and safe description of the program behavior.
2. It is automatic, i.e., the program does not have to be annotated.
3. It works for all programs in the chosen language.

The safe description means that calculated behavior includes all of the available result of real running. The approximate by abstraction of program behavior needs a price. The price to be paid is loss of information; the calculation will sometimes give only approximate information. Abstract interpretation using abstract domain generally has bounded values instead of concrete domain means unbounded real values [4, 5].

To apply abstract interpretation, we need to define a collecting semantics to consider properties of a subject to analyze. It is used for relatively complete and safe proof about considered properties. And we define a concrete semantics which describes all of available properties of program presented by concrete domain. Then, it needs to define an abstract semantics based on the concrete semantics to provide safe approximate information for program behavior while running [4].

A corresponding relationship between concrete semantics and abstract semantics are described as Galois connection that is defined by pairs of functions (α, γ) . Galois connection means that there is approximate abstract value corresponding with concrete value. For example we may represent the state of a program manipulating integer variable by ignoring the actual values of the variable and keeping their signs. Such abstraction may lose information by some operations on variable. A loss of precision may help making the semantics decidable.

Thus, abstract interpretation has been used to verify safety of software, to analyze loop bound for execution time through $\langle \text{interval, congruence} \rangle$ domain and to predict hardware behavior [6]. In our work, we use it to analysis Interval and to obtain analysis data of external inputs.

3 Method Detail

In this section, the analysis complex structural loops aiming at automation of static execution time analysis is discussed in detail.

There are two types of complex structural loops. First one uses a complex condition instead of simple increase/decrease. The second one uses a nested loop including branch inside. To analyze these two methods, there is a process of static execution time analysis for automation. The process of parametric WCET analysis is illustrated in Fig. 2.



Fig. 2. Process of parametric WCET analysis

First, it analysis the flows of target program and generates Control Flow Graph (CFG) in Control Flow & Structure Analysis. In Abstract Interpretation, it calculates interval domain of both index variables and related variables at each flow using states of control flow. Then Loop bound Analysis calculate loop bound of every path through information collected. Lastly, Timing Analysis constructs a formula for parametric execution time analysis as per both paths and loop bound. Timing Analysis is out of the scope of this paper.

3.1 Control Flow and Structure Analysis

The proposed method is based on a CFG. But there is some overhead to analyze external inputs on paths because the CFG has many path covering both feasible and unfeasible paths. Therefore we use the E-CFG to reduce overhead and to analyze each path inside compartmental loop as scopes [7]. Fig. 3 shows a program for example including nested loops.

```

int a, int b;
while(a < 30) {
    while(b < a) {
        if(b > 5)
            b = b * 3;
        else
            b = b + 2;
        if(b >= 10 && b <= 12)
            a = a + 10;
        else
            a = a + 1;
    }
    a = a + 2;
    b = b - 10;
}
    
```

Fig. 3. A sample code for example

An E-CFG of sample code presents a graph that has two sub-graphs as number of loops. Edges at the E-CFG indicate the state after performing vertices (nodes), so a series of edges is a path. Thus, when the sample has a set of feasible state $S = \{S_0, S_1, \dots, S_{12}\}$, it has a set of path : $P = \{P_1 = \{S_0, S_1\}, P_2 = \{S_2, S_3, S_4\}, P_3 = \{S_5, S_6, S_8, S_{10}, S_{12}\}, P_4 = \{S_5, S_6, S_8, S_{11}, S_{12}\}, P_5 = \{S_5, S_7, S_9, S_{10}, S_{12}\}, P_6 = \{S_5, S_7, S_9, S_{11}, S_{12}\}\}$.

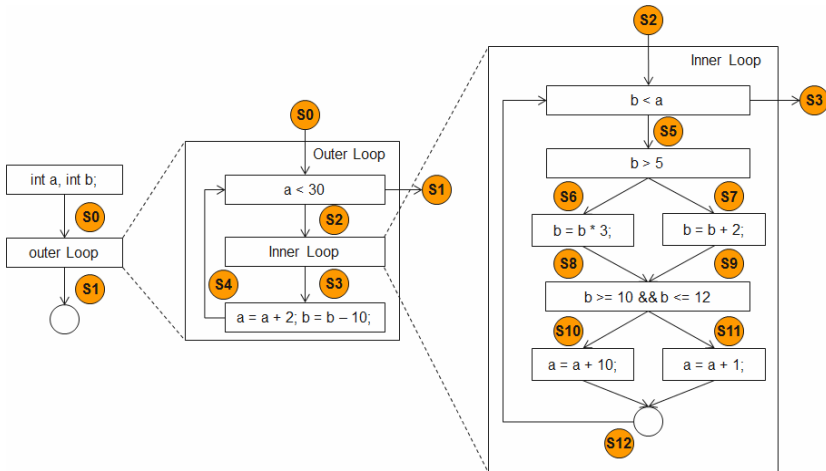


Fig. 4. An E-CFG of sample code

3.2 Abstract Interpretation

Constraints Analysis

There is a necessary condition to enter a particular flow. The condition calls a constraint. The constraint is located in selection or iteration statement and, is decided by control variables and constants in it. The constraint affects the decision for a state of variables in each path. So, it can serve to analyze interval of variables. There is a relationship of logical product among the constraints of scopes. For example, an innermost statement in nested selection could be executed when a state of the statement satisfies all conditions of each selection statement. And its state is equal to logical product of constraints.

In this paper, we use abstract domain as interval $[l ..u]$ to represent a state of each program point. Formally, constraints of a state is represented by function $C(S_n) = \{ (var, oper, const) \}$ where *var* is control variable and, *const* represents a constant operated with control variable or a computable result of operations. *oper* represents relational operator { " $>$ ", " $<$ ", " $=$ ", " \leq ", " \geq ", " \neq " } between *var* and *const*. Formula 1 shows states and constraints as results of analyzing sample in Fig. 2:

$$\begin{aligned}
 S_2 &= \{ [-\infty, 30], [-\infty, \infty] \}, C(S_2) = \{ (a, <, 30) \} \\
 S_5 &= \{ [-\infty, 30], [-\infty, \infty] \}, C(S_5) = \{ (b, <, a) \} \\
 S_6 &= \{ [-\infty, 30], [6, \infty] \}, C(S_6) = \{ (b, >, 5) \} \\
 S_7 &= \{ [-\infty, 30], [6, \infty] \}, C(S_7) = \{ (b, \leq, 5) \} \\
 S_{10} &= \{ [-\infty, 30], [6, \infty] \}, C(S_{10}) = \{ (b, \geq, 10), (b, \leq, 12) \} \\
 S_{11} &= \{ [-\infty, 30], [6, \infty] \}, C(S_{11}) = \{ (b, <, 10 \parallel b, >, 12) \}
 \end{aligned} \tag{1}$$

Feasible flow Analysis

As given above, it needs to satisfy constraint to enter a particular path. But if a result of logical product of constraints in a path is an empty set, the path is considered as an unfeasible path. Similarly, there are branches by conditions including index or external input in loops, and those can be unfeasible paths according to loop bound. So the feasible path is analyzed recursively according to scopes and constraints in Feasible flow analysis.

Therefore, a result of Feasible flow Analysis represents as constraint $C(P_n)$ when $P_n(1 \leq n \leq 5)$ in Formula 2:

As shown in Formula 2, constraint $C(P_3)$ and $C(P_5)$ are empty sets, there is no available value of variables a, b. Thus we can conclude that paths P_3 and P_5 are unfeasible paths.

External Input Analysis

A value of external input is unknown at a start state of target program. But, to reach a certain state, it needs to satisfy conditions, i.e. constraints, included until the state in flow. If there are two or more constrains to pass a particular state in a path, it must satisfy every constraint. External Input Analysis performs analysis using constraints from start state to end state of target program source. And, a result of logical product of constraints for each feasible path of program includes external variable then, it is available range of external input.

$$\begin{aligned}
& \dots \\
C(P_3) &= \{C(S_5) \cap C(S_6) \cap C(S_8) \cap C(S_{10}) \cap C(S_{12})\} \\
&= \left\{ (b, <, a), (b, >, 5), \left(b, \geq, \frac{10}{3}\right), \left(b, \leq, \frac{12}{3}\right) \right\} \\
&= \{\emptyset\} \\
& \dots \\
C(P_6) &= \{C(S_5) \cap C(S_7) \cap C(S_9) \cap C(S_{11}) \cap C(S_{12})\} \\
&= \{(b, <, a), (b, \leq, 5), (b, <, 10 - 2), (b, >, 12 - 2)\} \\
&= \{(b, <, a), (b, \leq, 5)\} \\
C(P_n) &= \begin{cases} C(P_1) = \{all\} \\ C(P_2) = \{(a, <, 30)\} \\ C(P_3) = \{\emptyset\} \\ C(P_4) = \{(b, <, a), (b, >, 5)\} \\ C(P_5) = \{\emptyset\} \\ C(P_6) = \{(b, <, a), (b, \leq, 5)\} \end{cases} \quad (2)
\end{aligned}$$

First, it classifies each feasible paths of program by scope. Those are results of classification from sample: $\{P_1\}$, $\{P_2\}$, $\{P_3, P_4, P_5, P_6\}$. Second, it uses Cartesian product to find full path and, calculates constraint as Formula 3.

$$\begin{aligned}
C(\{P_1, P_2, P_3\}) &= \{\emptyset\} \\
C(\{P_1, P_2, P_4\}) &= \{(a, <, 30), (b, <, a), (b, >, 5)\} \\
C(\{P_1, P_2, P_5\}) &= \{\emptyset\} \\
C(\{P_1, P_2, P_6\}) &= \{(a, <, 30), (b, <, a), (b, \leq, 5)\}
\end{aligned} \quad (3)$$

As shown in Formula 3, there are two feasible paths with constraints. So, we know that external input variable a and b have interval $[-, \dots, 29]$. Particularly when variable b has interval $[6., 28]$, we know that a path $\{P_1, P_2, P_4\}$ must be executed.

3.3 Loop Bound Analysis

Compute loop bound and Control variable information table

In current researches, loop bound analysis has aimed at analyzing simple loop just using regularly increasing index. But there is complex loop that has variable condition as branch shown as Fig. 3. In this instance, it is hard to analyze loop bound using <interval, congruence> domain in existing manner. Accordingly, in this paper, we use a formula to compute loop bound and analyze loop bound separately by the feasible path. The formula considers a case that increasing condition is not only an arithmetical sequence but also a geometric sequence. So it uses $i_{n+1} = (k * i_n) + l$ ($n > 0$) as an increment to reflect variable index variable. The loop bound calculation formula which is applied increment of index is as follow:

$$I = \begin{cases} \left\lceil \log_k \left(\frac{(k-1)(x_n - x_0)}{(k-1)x_0 + l} + 1 \right) \right\rceil, & k \neq 1 \\ \left\lceil \frac{x_n - x_0}{l} \right\rceil, & k = 1 \end{cases} \quad (4)$$

We also use control variable information table that analyzes complex operation of condition to apply Formula 4. It can analyze control variables that are criterion of iteration, operators and increments as scopes.

Table 1. The sample of Control variable information table

Scope ID	Constraint	Condition	Increase	
			k	l
1		a < 30		
2		b < a		
2	b > 5	b	3	
2		a		1
2	b ≤ 5	b		2
2		a		1
1		a		2
1		b		-10

The first column in the table is a scope of loop to apply Formula 4. The Constraint means a branch in the loop and represents an available range of control variable with condition as the above row. The Condition is criterion of iteration, even though it is only name of variable, it means i_n , control variable, for the Increase operation condition. The ‘k’ and ‘l’ in the Increase column are same in ‘k’ and ‘l’ of $i_{n+1} = (k * i_n) + l$ ($n > 0$) respectively to calculate index.

Analyzing complex loop

A complex loop indicates nested loop that has two or more control variable. To analyze the complex loop, it is classified according to dependency by control variable into three types: Outer loop depending on (or it can be ‘dependent’) inner loop, inner loop depending on (or it can be ‘dependent’) outer loop, and interdependence. According to the classification, dependent loop bound is computed first, and then the others are done as applying its variation. To analyze third type that is most difficult, it needs to count states before entering inner loop in outer loop. In other words, inner loop bound is calculated using loop bound calculation formula, and outer loop bound is calculated by counting number of states applied as a result of computing inner loop bound.

There is a Formula 5 to compute loop bound of inner loop using parametric manner as each constraints with paths. And a result of loop bound calculated from aforementioned rule and Formula 5 is shown in Table 2.

$$I_{inner} = \begin{cases} P_4: \left\lceil \log_3 \left(\frac{2(a+b)}{2b+1} \right) \right\rceil, a = [7..30], b = [6..29], b < a \\ P_6: |a - b - 1|, a = [-\infty..30], b = [-\infty..5], b < a \end{cases} \tag{5}$$

Table 2. The result of loop bound calculation

Outer loop index	value		Inner loop bound
	a	b	
0	1	1	0
1	3	-9	9
2	14	11	1
3	17	23	0
4	19	13	1
5	22	29	0
6	24	19	1
7	27	47	0
8	29	37	0
9	31	27	0

As a guide, Inner loop bound 9 is sum of 7, ($b \leq 5$) and 2, ($b > 5$) at outer loop index 1 of Table 2.

4 Evaluation

We have compared our analysis with previous analysis [16] using the Mälardalen WCET Benchmark suite [19] in this section. A main indication for comparison is the percentage of analyzing loop bound, which is the largest part of execution time analysis. Table 3 gives results of loop bound analysis for comparison. And expressions of Table 3 follow [16]. The column (Loops) gives the number of loops in program and the number (#B) and the percentage (%B) gives loop bound by analysis. It also gives the number (#E) and the percentage (%E) of loops which are exactly bound.

Table 3. The result of loop bound analysis compare with previous research using AI

Program	Loops	Previous analysis				Our analysis			
		#B	%B	#E	%E	#B	%B	#E	%E
Bs	1	0	0%	0	0%	1	100%	1	100%
duff	2	1	50%	1	50%	2	100%	2	100%
edn	12	12	100%	9	75%	12	100%	12	100%
fft1	30	7	23%	3	10%	24	80%	23	77%
fir	2	2	100%	1	50%	1	50%	1	50%
jcomplex	2	0	0%	0	0%	2	100%	2	100%
ludcmp	11	6	55%	5	45%	11	100%	8	73%
ns	4	1	25%	1	25%	4	100%	4	100%
prime	2	0	0%	0	0%	1	100%	0	0%
qurt	3	1	33%	1	33%	3	100%	0	0%
Total	69	30	39%	21	29%	61	93%	53	70%

The Total row summarizes comparison of our analysis results with previous analysis. We see that analyzed loop bounds of our analysis increase by 93% when

previous one is 39% and exact loop bounds increase by 70% when previous one is 29%. There is great difference by ratio because we used programs that were difficult to analysis and they had lower analysis ratio.

Consequently, it shows that our proposal contributes to analyze the programs using complex loop or floating point index which could not be analyzed before.

5 Conclusion

Existing parametric WCET analysis had problems in limited loop analysis supporting simple loop bound. It analyzes simple increase/decrease of integer type index and loop bound that is not related to inside flow of loops. For this reason, it needs a measure to analyze varied structural loops and detailed loop bound applying internal feasible paths of loops.

In this paper we present automatic analysis of parametric static execution time for complex loop contained external input and feasible path in loop as variable conditions.

The proposed method can analyze complex structural loop that has nested condition through Control variable information table and feasible path as loop bound. This can analyze complex structured program that could not be analyzed before in existing parametric WCET. It reduces user input and helps to get accuracy and tight execution time using detailed information of feasible paths and loop bounds.

In future, we plan to improve performance and size of the table. The table has some problems that are caused by increasing number of states. We also plan to extend the approach to evaluate the dangers of violating deadline to use static execution time.

Acknowledgments. This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency) (NIPA-2011-C1090-1131-0003), and supported by R&DB Support Center of Seoul Development Institute, Korea, under Seoul R&BD Program(ST100107).

References

1. Laplante, P.A.: Real-Time System Design and Analysis. IEEE Press, Los Alamitos (2004)
2. Wilhelm, R., Mitra, T., Mueller, F., Puaut, I., Puschner, P., Staschulat, J., Stenström, P., Engblom, J., Ermedahl, A., Holsti, N., Thesing, S., Whalley, D., Bernat, G., Ferdinand, C., Heckmann, R.: The worst-case execution time problem - overview of methods and survey of tools. *ACM Transactions on Embedded Computing Systems(TECS)* 7(3), 1–53 (2008)
3. IEEE Standard Glossary of Software Engineering Terminology, IEEE Std 610.12- (September 1990)
4. Cousot, P., Cousot, R.: Abstract Interpretation:a Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints. In: *Proceedings of ACM Symp. on Principle of Programming Languages*, pp. 238–252. ACM Press, NY (1977)

5. Cousot, P., Cousot, R.: Systematic Design of Program Analysis Frameworks. In: Proceedings of ACM Symp. on Principle of Programming Languages, pp. 269–282. ACM Press, NY (1979)
6. Blanchet, B., Cousot, P., Cousot, R., Feret, J., Mauborgne, L., Mine, A., Monniaux, D., Rival, X.: A Static Analyzer for Large Safety-Critical Software. In: Proceedings of ACM SIGPLAN conference on Programming Language Design and Implementation (PLDI), pp. 196–207 (2003)
7. Kim, Y.-k., Shin, W., Kim, T.-w., Chang, C.-h.: A Methodology of Analyzing External Input for Improving Flexibility of Static Execution Time Analysis. In: Proceedings of Annual International Conference on Software Engineering, pp. 67–72 (2010)
8. Nielson, F., Nielson, H.R., Hankin, C.: Principles of Program Analysis. Springer-Verlag New York, Inc. Secaucus, NJ (1999)
9. Kirner, R., Puschner, P.: Discussion of Misconceptions about WCET Analysis. In: Proceedings of WCET Workshop 2003, pp.61-64 (2003)
10. Tidorum Ltd., Bound-T Time and Stack Analyser,
<http://www.tidorum.fi/bound-t/>
11. Puschner, P.: Is Worst-Case Execution-Time Analysis a Non-Problem? - Towards New Software and Hardware Architectures. In: Proceedings of Second Euromicro International Workshop on WCET Analysis, Technical Report, York YO10 5DD, United Kingdom, Department of Computer Science, University of York (2002)
12. Puschner, P.: Algorithms for Dependable Hard Real-Time Systems. In: Proceedings of 8th IEEE International Workshop on Object-Oriented Real-Time Dependable Systems (WORDS), pp. 26–31 (2003)
13. Gustafsson, J., Lisper, B., Kirner, R., Puschner, P.: Code Analysis for Temporal Predictability. *Real-Time Systems* 32(3), 253–277 (2006)
14. Kirner, R., Knoop, J., Prantl, A., Schordan, M.: WCET Analysis: The Annotation Language Challenge. In: Proceedings of 7th International Workshop on Worst-Case Execution Time Analysis (2007)
15. Kirner, R., Kadlec, A., Puschner, P., Prantl, A., Schordan, M., Knoop, J.: Towards a Common WCET Annotation Language: Essential Ingredients. In: Proceedings of 8th International Workshop on Worst-Case Execution Time Analysis, WCET (2008)
16. Bygde, S.: Static Analysis based on Abstract Interpretation and Counting of Elements. Ph.D. thesis Malardalen Univ., Sweden (2010)
17. Ermedahl, A., Sandberg, C., Gustafsson, J., Bygde, S., Lisper, B.: Loop Bound Analysis based on a Combination of Program Slicing, Abstract Interpretation, and Invariant Analysis. In: Proceedings of 7th International Workshop on Worst-Case Execution Time Analysis, WCET (2007)
18. Kim, Y.-k., Shin, W., Kim, T.-w., Chang, C.-h.: Design and Implementation of PS-Block Timing Model Using PS-Block Structue. *The KIPS Transactions* 13-D(4), 613–618 (2006)
19. Mälardalen University. WCET project homepage (2007),
<http://www.mrtc.mdh.se/projects/wcet>

Software Reuse: MDA-Based Ontology Development to Support Data Access over Legacy Applications

Heru-Agus Santoso^{1,2}, Su-Cheng Haw², and Chien-Sing Lee²

¹ Faculty of Computer Science, Dian Nuswantoro University, 50131 Semarang - Indonesia

² Faculty of Information Technology, Multimedia University,
63100 Cyberjaya - Malaysia Indonesia

{heru.agus.santoso08, schaw, cslee}@mmu.edu.my

Abstract. Unified Modeling Language (UML) and ontology share common properties such as classes, properties and instances. We propose using Model-Driven Architecture (MDA) enriched with ontological approach to provide ontology development method. The method leverages the UML model in the initial phase of ontology development, and then the produced ontology is aligned with specific domain ontology. The steps involved consist of: (1) generating the UML model from the legacy application, (2) generating OWL ontology from the UML model, (3) enriching the generated ontology with domain ontology, and (4) incorporating the ontology in ontology-based query answering. For simulation, the query is implemented using SPARQL over the OpenBiblio database.

Keywords: software reuse, MDA-based ontology development.

1 Introduction

In software reuse, reusable asset, i.e. reusable software and reusable knowledge can be used to improve software quality and productivity [1]. To enable reuse, it is important to use common terms and vocabularies, to increase interoperability among applications. Such common terms and vocabularies are found in models. In this paper, we refer to Model-Driven Architecture (MDA) as our model. MDA is a modeling standard that provides a viable solution to represent the system as a model, and also to represent a model in another model [2, 3]. We choose MDA as our model because its main goal is to achieve better interoperability, portability and integration among the systems [3]. Technically, interoperability and integration can be achieved by providing a facility for resource sharing. Meanwhile, portability focuses on the ability of a system to reuse the existing resource when it migrates into another environment.

Taking the software product line (SPL) paradigm [4], in an open source community, a single application downloaded by a user can be seen as single variant of the product. The problem arises when the user desires to use new specific features which are not available in the repository. Some users will then refer to the open source community. In the open source community, it is possible to take new features from an arbitrary repository, or even create a new feature and integrate it with the application. For example, PHP BB (discussion forum), Compiere ERP+CRM

(Enterprise Resource Planning & Customer Relationship Management system), PHP Fusion (Content Management System), Mybloggie (blogging system) and OpenBiblio (library system automation) have been downloaded 2,734,846, 871,778, 214,505, 21,372 and 97,898 times respectively, as of August, 2010.

Furthermore, taking OpenBiblio as an example, OpenBiblio is OSS for a library automation system written in PHP programming language and operates using a standard relational data model. Many users are still using the earlier version of the application created in 2002. We have investigated; that each version of OpenBiblio has different database schema. This kind of isolated data will be more useful if it can be interlinked or enhanced with the rich data available in the Web. However, it is common for legacy systems to have little relation with the current technology. Therefore, an approach dealing with reusability while keeping its compatibility is important. The main contribution of this paper is we propose a hybrid approach of ontology development, combining MDA-based and ontological approach in the context of software reuse.

In the Section 2, we present our rationale for using MDA-based ontology to enable reusability while maintaining compatibility. Section 3 presents the preliminaries of the approach. Section 4 discusses our proposed framework. The conclusion is presented in Section 5.

2 Rationale for an MDA-Based Ontological Approach

The main idea of ontology is to enable knowledge sharing [5]. Dealing with knowledge sharing and reuse, the World Wide Web Consortium (W3C) recommended the use of standard ontology languages such as Resources Description Framework/Resource Description Framework Schema (RDF/RDFS) and Web Ontology Language (OWL/OWL 2) to provide the “executable” model of a domain of interest [6].

One of the advantages of adopting ontology in the software development process is to enable the people involved in the development to use common vocabularies, thereby increasing understanding among them [7]. Gašević et al. [8] developed ontology using a software engineering approach. They found some closely related elements among UML and ontology, such as classes, properties, and inheritance. Furthermore, the UML class diagram is an adequate source for ontology development. This claim is reinforced by Calvanese et.al [9], in which their works are centered around Description Logics (DLs). They stated that DL-lite_A (one of the family member of DLs- the logical formalism for Semantic Web) can capture knowledge from the UML class diagram.

The framework presented in this paper is mostly inspired by MDA-related work such as in [2] and [10]. Since the functionality of the generated ontology is to support ontology-based data access, we also take into account the information from database. For this study, we reused the modeling artifact of OpenBiblio¹ to automatically generate the UML class diagram, then build a Web ontology on top of the model. The

¹ <http://obiblio.sourceforge.net/>

generated ontology is then aligned with the related domain ontology. Next, it is incorporated in query answering, supporting ontology-based data access over relational data. In providing data access support using an ontological approach, once the dataset of the application is represented as an intelligent view, it can then provide a wider interpretation supporting integration. The data can also be annotated without concern about their physical structure.

3 Preliminaries

3.1 Ontology Definition in Semantic Web Context

The most cited definition of ontology in the Semantic Web community is provided by Gruber [5]. He defined ontology as an “explicit specification of a conceptualization”. Through conceptualization, implicit knowledge of domain becomes explicit. It can be used as a common vocabulary and shareable for different applications and organizations. Davies et.al [10] classified ontologies based on their intended purpose, i.e. (1) upper-level ontology, (2) domain ontology, and (3) application ontology. The upper-level ontology is the most general type of ontology. Examples are Dublin Core and Suggested Upper Merged Ontology (SUMO) widely used in a large variety of application areas. Domain ontology represents the knowledge of a specific domain, such as Gene ontology and Amino Acid ontology. Application ontology, however, is the ontology used in a specific application.

The concepts, roles, instances and axioms used in the conceptualization should be explicitly defined by means of representing them using formal languages. The aim of representing ontology using formal language is that arbitrary application can process the meaning of information instead of just displaying. OWL uses Description Logics (DL) as the foundation for logic. OWL formally describes knowledge of a specific domain using classes, properties and their relationships. Fig. 1 depicts the example of RDFS/OWL for the Biblio Ontology². W3C recommends OWL 2 as an extension and revision of OWL in October 2009. In OWL 2, scalability of the OWL 2 fragments are addressed in terms of profiles. There are three OWL 2 profiles, which are OWL 2 EL, OWL 2 QL and OWL 2 RL.



Fig. 1. Graphical representation of Biblio ontology
(loaded using Protégé 4.1, URL: <http://purl.org/ontology/bibo>)

² <http://bibliontology.com>

OWL as an ontology language allows automatic inferencing or reasoning over a dataset aligned to them [10]. The functionality of ontology explored in this paper is how to use ontology for query answering over relational data. Once the dataset is represented as an intelligent view, it can then provide a wider interpretation of the data in which we can improve interoperability and usage of them.

3.2 MDA-Based Ontology Infrastructure

MDA is a modeling approach maintained by the Object Management Group³ (OMG). In software development, MDA first models an application using a modeling language which is based on Meta Object Facility (MOF). In providing MDA-based ontology metamodeling, MOF is used to define ontology metamodel based on the ontology language, i.e. OWL. Based on [8], the MDA-based ontology metamodeling infrastructure consists of the following elements:

1. MOF is used to provide the ontology metamodel.
2. The ontology metamodel is designed by enclosing the common concepts of ontology based on OWL.
3. UML model and ontology metamodel are serialized using XML Metadata Interchange (XMI). Thus, the transformation from UML model to OWL ontology can be done using Extensible Stylesheet Language Transformation (XSLT).

In the MDA architecture such as shown in Fig. 2, MOF and RDF/RDFS are located in meta-metamodel layers. The relationship between MOF and RDF/RDFS constructs are briefly presented in Table 1 [2, 10].

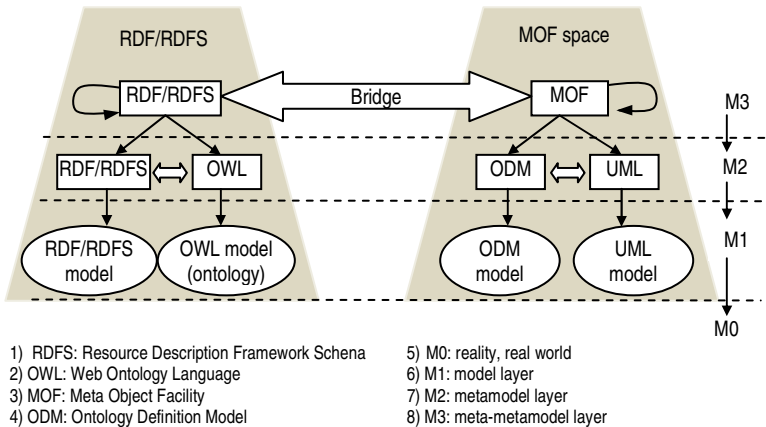


Fig. 2. Model Transformation between RDF/RDFS and MOF

³ <http://www.omg.com>

Table 1. Relationship between MOF and RDF/RDFS

MOF	RDF/RDFS	Description
ModelElement	rdfs:Resource	ModelElement is root element in MOF, provides atomic construct of model, where rdfs:Resource represents all resource used by RDF constructs
DataType	rdfs:Datatype	Both DataType and rdfs:Datatype are used to provide primitive data type
Class	rdfs:Class	Class in MOF defines group of objects, in RDF/RDFS defines group of resources or individuals.
Association	rdf:Property, rdfs:Domain and rdfs:Range	Association expresses relationship among MOF metamodel elements, in RDF/RDFS, rdf:Property, rdfs:Domain and rdfs:Range define relationship among resources or individuals from different classes.
Attribute	rdfs:Resource	Attribute in MOF is related to rdfs:Resource

3.3 Model Transformation

Model transformation is an automatic generation process from one model to another model based on a transformation definition, expressed using model transformation language [11]. The transformation definition is implemented in terms of a set of rules that match elements from the source model to specific elements of the target model. Model transformation involves source model, target model, rules and transformation language. According to Gašević et al. [2], the classification of transformation language is defined as three types, which are:

1. Declarative language: the transformation language which specifies the relationship between source and target model applied without describing execution order.
2. Imperative language: the transformation language which specifies the sequence of the steps to get results explicitly.
3. Hybrid: mix between declarative and imperative language.

To deal with model transformation, *modeling space* helps software developers to address their problem with a right approach, i.e. using a precise (meta)modeling language [2]. One of the advantages of using RDF/RDFS is one can define a model in a computer-processable way. Fig. 2 shows an excerpt of the transformation model between MOF modeling space and RDF/RDFS modeling space, adapted from [2].

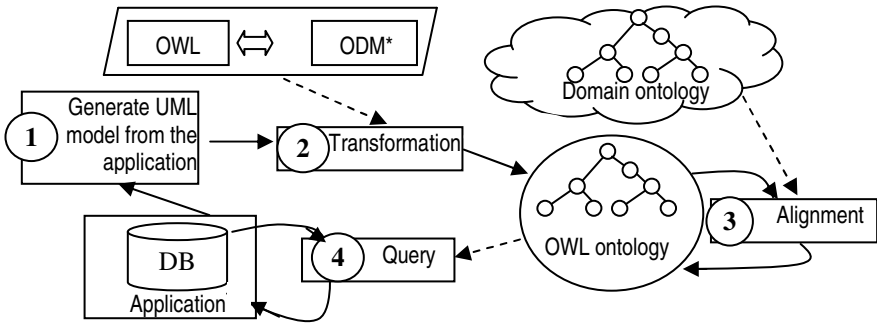
The transformation bridge between RDF/RDFS and MOF is established at the M3 layer. Since MOF provides minimal requirements to define another modeling language, the transformation from RDF/RDFS to MOF, and vice versa aims to support model transformation based on meta-metamodel.

4 Our Proposed Framework

In an effort to provide query answering to support ontology-based data access, we identified four main steps as follows:

1. Generate UML model, specifically UML class diagram.
2. Generate OWL ontology from the model.
3. Enrich the generated ontology (application ontology) with related domain ontology.
4. Incorporate the ontology in query answering using SPARQL. SPARQL is the RDF query language which officially announced by W3C as query language recommendation in October, 2008.

Fig. 3 depicts the framework of our approach.



*: ODM defines ontology metamodel used as semantic foundation of UML model [2]
 ---> : reference

Fig. 3. Our proposed framework for supporting data access over a legacy application

4.1 Generating UML Model

Generally, a model describing a particular system can provide a set of semantics which may automatically be transformed into another model. In software design approach, the transformation model in MDA is usually used for domain engineering and for providing system functionality in platform-independent representation.

Since our approach aims to reuse the modeling aspect of the legacy application, model transformation approach is one of the solutions to directly generate UML class and object diagram. Several tools either licensed or free are available on the Web, such as Sparx Systems Enterprise Architect⁴ (licensed), PHP_UML⁵ and BOUML⁶ (both free). By using BOUML, we can generate 88 classes, 779 attributes and 38 relations from source code of OpenBiblio. The generated UML class diagram components then should be selected and refined, to produce a proper class diagram. Fig. 4 depicts the generated UML class diagram from PHP source code. Note that, some attributes and operations within each class are invisible due to space limitation.

UML object diagram can be seen as an instance of the UML class diagram. It provides example of concrete objects and their relationships. In deriving a UML

⁴ <http://www.sparxsystems.com.au/products/ea/index.html>

⁵ http://pear.php.net/package/PHP_UML/

⁶ <http://bouml.free.fr/>

object diagram from a UML class diagram, we also need to leverage the database information, to identify which instances and relation are important regarding the functionality of ontology.

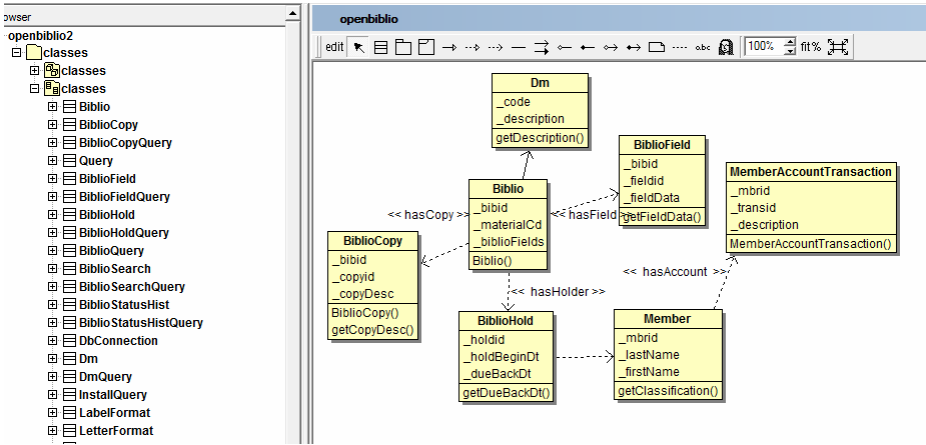


Fig. 4. UML class diagram generated from PHP source code

4.2 Generating Ontology from UML Model

The relevant UML diagrams regarding OWL ontology engineering based on [2] are UML class and object diagram. The UML class diagram can be used to create the ontology schema, in which we can generate class relationship and class taxonomy, whereas UML object diagram can be used to model the instances of the ontology. We consider the objects - regardless their state and behavior used in object-oriented modeling paradigm, which can be seen as the instances of the ontology, are stored in relational data.

UML model which is based on object-oriented paradigm is limited when applied to ontology development [8]. A common limitation of this kind of approach is that the generated ontology is still an initial version. Thereby it needs some refinements and enrichments using ontological approach, e.g. ontology mapping or ontology alignment.

Before further discussion on how to generate ontology from a UML model, let us look at the definition of ontology. Generally, ontology can be defined as 4-tuples:

$O = (C, R, I, A)$, where:

- C: set of concepts or classes c_i , i.e. {Book, Author, ...};
- R: set of roles or properties r_i , i.e. {hasClassification, hasAuthor, ...};
- I: set of instances or individuals i_i , i.e. {John Davies, Diego Calvanese, ...};
- A: set of axioms a_i , i.e. {John Davies is an author, ...};

Based on [2, 8], the relationship between the ontology elements and UML model is presented in Table 2.

Table 2. Ontology and UML relationship

Components	OWL ontology	UML	Description
Concept or class	Concept or class is a set of individuals or instances	Class is a set of objects	In ontology, classes are related to each other, like a set of classes represented in UML class diagram.
Role or association	Role is implemented using ObjectProperty with its restriction, to connect individuals from two different classes	Association is used to connect objects of two (or more) classes	ObjectProperty can be functional, inverse functional, transitive, symmetric property, etc. Its restrictions can be existential or universal quantification, cardinality, etc. In UML, properties of association can be attribute, operations, etc.
DataTypeProperty or attribute	DataTypeProperty has two main elements: <i>domain</i> refers to as its class domain and <i>range</i> , refers to XML Schema Datatype (XSD).	Attribute is defined locally in a class	ObjectProperty and DataTypeProperty of ontology can be associated as class stereotype <<ObjectProperty>> and <<DataType Property>> in class diagram
Individual or instance	An individual may belong to more than one classes	Each instance belongs to one class	In UML model, instance is regarded as UML object
Axiom	Consists of axiom about class, object or data property, assertion, annotation and so on	In class diagram, generalization-specification and disjointness can be represented as axiom	The main role of axiom is used to describe the relationship among concepts or class

As depicted in Fig. 3, the transformation model between RDF/RDFS and MOF is established at the M3 layer. The consequence is that, the transformation process is supposed to do carry out a meta-metamodel to meta-metamodel transformation with regards to the four-layers in the MDA architecture. RDF/RDFS itself can be a metamodel or model, but MOF is only possible as a meta-metamodel interface in this setting. MOF's idea is to serve as the framework for integrating metadata in a platform independent manner. MOF's minimal requirements to define another modeling language are: [12]: (1) Class - to model entities, e.g., UML class, association; (2) association - to represent relationship among classes; (3) data type, and (4) package, i.e. group of concept.

We use the direct transformation model as depicted in Fig. 5. As shown in the figure, XSLT functions as a template, which consists of a set of rules. It is used to match XMI constructs, and transform the matched constructs to related OWL primitives. Next, the result can be loaded and refined using an ontology editor such as Protégé.

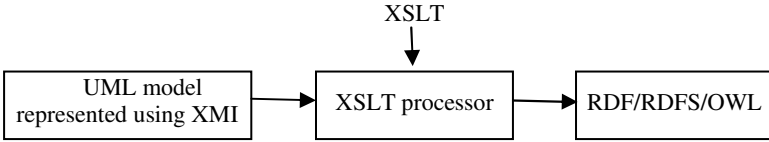


Fig. 5. MDA-based transformation using XMI and XSLT

4.3 Enriching the Generated Ontology

Enriching the generated ontology or application ontology with domain ontology can be achieved using ontology alignment. Ontology alignment, which is also known as ontology matching, is the process of finding a set of relations between two different ontology entities [13]. Several alignment systems are available, such as those hosted at the Ontology Matching website⁷. One of them is Prompt, a plugin of the Protégé ontology editor for ontology matching and merging. Generally, those systems cannot produce the matched ontological component automatically. It needs a semi-automatic approach.

The goal of alignment is to identify some related vocabulary entities. One of the approaches to determine the relation between two ontologies is Relation Overlap (RO). RO defines the relation accuracy based on the mean value of similarity between domain and range of class [14]. The most common semantic relation between concepts from different ontology are class equivalence (\equiv) and class subsumption (\subseteq) [15]. Equivalence states that there are equivalence aspects in two different ontologies regarding some specific criteria. Equivalence can be equality if two classes are strongly related and it represents exactly the same class. Subsumption states that one class represents more specific aspects than the other class, and vice-versa. Class subsumption also refers to sub-class and super-class relation.

Let O_1 be an application ontology; O_2 be a domain ontology; c_iO_1, c_iO_2 be a class from O_1 and O_2 respectively; $C_EQU()$ be a predicate to determine the class equivalence from two ontologies; $C_SUB()$ be a predicate to determine the subsumption relation between two classes; $Sim()$ is a predicate to obtain concept or property similarity; and $DP()$ is predicate to obtain a set of data properties or attributes from a concept. The equivalencies and the subsumption relation of O_1 and O_2 can be defined as follows:

a) Class equivalence

$$C_EQU(c_iO_1, c_iO_2) \rightarrow (c_iO_1 \equiv c_iO_2) \vee (Sim(c_iO_1, c_iO_2) > similarity_threshold) \vee (Sim(DP(c_iO_1), DP(c_iO_2)) > similarity_threshold), \text{ for } i=1,2,\dots,n.$$

c_iO_1 is said equivalent to c_iO_2 if it satisfies at least one of the following requirements:

1. c_iO_1 and c_iO_2 are semantically equivalent, i.e. Book=Book, or
2. Similarity between the name of c_iO_1 and c_iO_2 > similarity threshold, or
3. They have similar data properties or attributes.

⁷ <http://www.ontologymatching.org/projects.html>

b) Class subsumption

$$C_SUB(c_iO_1, c_iO_2) \rightarrow (c_iO_1 \subseteq c_iO_2), \text{ for } i=1,2,\dots,n.$$

c_iO_1 is said as sub class of c_iO_2 iff $c_iO_1 \subseteq c_iO_2$. c_iO_1 is subsumed by c_iO_2 iff every instance of c_iO_1 is also instance of c_iO_2 .

Fig. 6. shows an excerpt of two ontologies, the generated ontology and domain ontology. The possible related vocabulary entities are author subsumed by Person, Material_type_dm (equivalent class with Document), and Collection_dm (also equivalent class with Collection).

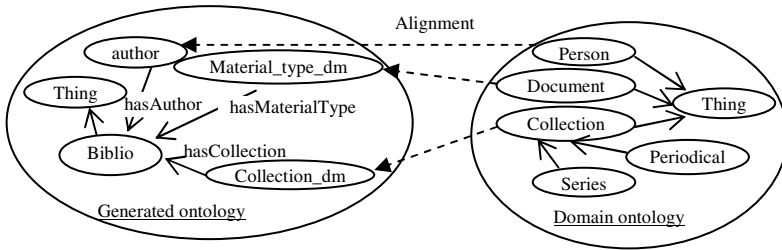


Fig. 6. Alignment of the generated ontology with domain ontology

4.4 Ontology-Based Data Access

This section briefly describes the implementation of ontology-based data access using conjunctive query (CQ). The basic form of CQ formulated over an ontology is defined as follow [16]:

$$q(a) \leftarrow \bigwedge_i^n c_i(a,b), i \in n, i \geq 1; \text{ where:}$$

- $q(a)$: the head of conjunctive query
- a, b : variable or constant involving concept, role, value domain or attribute. a is usually called distinguished variable.
- $c_i(a,b)$: the body of conjunctive query, it may employ non-distinguished variable.

The query is posed in terms of ontology-based data access iff $c_i(a,b)$ (as a set of conjuncts of the body) using concepts and roles name occurring in the corresponding ontology. The answer of $q(a)$ is a set of tuples which satisfy $\exists b.c_i(a,b)$ in the database.

For a given query:

“Who are the author(s) for books classified in ‘Family and Community Nursing’ and published in year 2000”.

Several facts derived from the query to form the triples graph are:

1. Author wrote book
2. Book has year of publication
3. Each book is classified
4. Family and Community Nursing is a book classification

As explained in the previous section, OWL uses DL as the foundation for logics. In the context of DL, ontology consists of set of terminological axiom (TBox) and assertional knowledge (ABox) of the form $O = \langle T, A \rangle$ [17]. The terminological axiom T of the above facts which represents the concept relationships in this domain are: (1) $\text{Biblio} \sqsubseteq \exists \text{hasAuthor}$, (2) $\text{Biblio} \sqsubseteq \exists \text{hasYear}$, and $\text{Biblio} \sqsubseteq \exists \text{hasDescription}$, whereas “Family and Community Nursing is a book classification” is an example of assertional knowledge A . Assertional knowledge describes the objects within specific domain. Fig. 7 illustrates the graph structure related to the above facts:

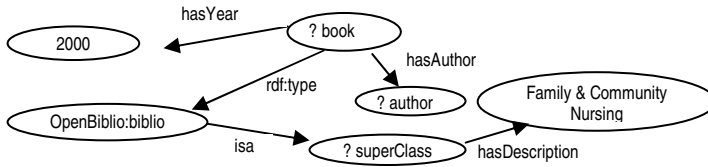


Fig. 7. RDF Triples

Given an ontology O and a query CQ . O is satisfiable after further checking using **HermiT⁸** reasoner. Since O is satisfiable, then the CQ evaluation over $O = \langle T, A \rangle$ is performed by computing certain answers of the CQ . The answer is set of individuals as presented in Fig. 8. In this example, the query is implemented using SPARQL over the **OpenBiblio** database.

Query

```
PREFIX rdfs:
<http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-
rdf-syntax-ns#>
PREFIX OpenBiblio:
<http://localhost:2020/biblio#>

SELECT ?author ?title WHERE
{ ?book OpenBiblio:year "2000" .
  ?book OpenBiblio:author ?author .
  ?book OpenBiblio:title ?title .
  ?book OpenBiblio:collection_cd ?type .
  ?book rdf:type ?className .
  ?className rdfs:subClassOf ?superClass .
  ?item rdf:type ?superClass .
  ?item OpenBiblio:code ?type .
  ?item OpenBiblio:description "Family &
Community Nursing" . }
```

Answer:

Author	Book
Clark, marry	Handbook community health nursing
Elizabeth T.Anderso	Community As Partner Theory and practice in Nursing
Cash, Jill C	Family practice guidelines
Jeffrerson, Tom	Elementary Economic Evaluation in Health Care, Second Editions
Green, Laurence W	health promotion Planning An Educational and Enviromental Approach, 2nd ed

Fig. 8. Query implementation and the answer

⁸ <http://hermit-reasoner.com>

5 Conclusion

We have successfully used MDA to provide the framework to represent a system as a model. We have also used UML to model the OWL ontology. The relevant UML diagrams regarding OWL ontology engineering are class and object diagrams. UML class diagram is used to generate class relationships while the UML object diagram can be used to generate concepts and instances of the ontology. However, it has limitations since the generated ontology is still an initial version of the ontology. Therefore, it needs some refinements and enrichments. The refinement can be done by using the ontological approach such as ontology matching or ontology alignment. Ontology alignment is done to identify some related vocabulary entities such as class equivalence and class subsumption regarding the related domain ontology. Next, the ontology is incorporated in query answering using SPARQL over the legacy database.

References

1. Frakes, W.J., Kang, K.: Software Reuse Research: Status and Future. *IEEE Transactions on Software Engineering* 31(7) (2006)
2. Gašević, D., Djuric, D., Devedžić, V.: *Model Driven Engineering and Ontology Development*, 2nd edn. Springer, Heidelberg (2009)
3. Mellor, S.J., Scott, K., Uhl, A., Weise, D.: Model-Driven Architecture. In: Bruel, J.-M., Bellahsene, Z. (eds.) *OOIS 2002*. LNCS, vol. 2426, pp. 290–297. Springer, Heidelberg (2002)
4. Der Linden, F.V., Lundel, B., Chastek, G.: Open Source Software Product Lines. In: *12th International Software Product Line Conference – SPLC 2008*, pp. 387–387. Limerick, Ireland (2008)
5. Gruber, T.R.: Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition* 5(2), 199–220 (1993)
6. Savo, D.F., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M., Romagoli, V., Ruzzi, M., Stella, G.: MASTRO at Work: Experience on Ontology-based Data Access. In: *Proceeding 23rd Workshop on Description Logics (DL2010)*, CEUR-WS 573, Waterloo, Canada (2010)
7. Falbo, R.A., Guizzardi, G., Duarte, K.C., Natali, A.C.: Developing Software for and with Reuse: An Ontological Approach. *CSITeA*, pp. 311 – 316, Brazil (2002)
8. Gašević, D., Djuric, D., Devedžić, V.: MDA-based Automatic OWL Ontology Development. *International Journal on Software Tools and Technology Transfer* 9, 103–117 (2007)
9. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Conceptual Modeling for Data Integration. In: Borgida, A.T., Chaudhri, V.K., Giorgini, P., Yu, E.S. (eds.) *Conceptual Modeling: Foundations and Applications*. LNCS, vol. 5600, pp. 173–197. Springer, Heidelberg (2009)
10. Davies, J., Studer, R., Warren, P.: *Semantic Web Technology*. John Wiley and Sons Ltd., West Sussex (2006)
11. Kurtev, I.: *Adaptability of Model Transformations*. PhD thesis, University of Twente, CTIT Ph.D.-thesis series No. 05-71 (2005)
12. Kelly, S., Tolvanen, J.: *Domain Specific Modeling: Enabling Full Code Generation*. John Wiley and Sons Inc., Hoboken (2008)

13. Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D., Antoniou, G.: *Ontology Change: Classification and Survei*. In: *The Knowledge Engineering Review*, vol. 00(01-29), pp. 1–19. Cambridge University Press, Cambridge (2007)
14. Maedche, A., Staab, S.: *Ontology Learning for The Semantic Web*. *IEEE Intelligent Systems and Their Applications* 16(2), 72–79 (2005), ISSN: 1541-1672
15. Davies, J., Groblenik, M., Mladenic, D.: *Semantic Knowledge Management: Integrating Ontology Management, Knowledge Discovery, and Human Language Technology*. Springer, Heidelberg (2009)
16. Stamou, G., Trivela, D., Chortaras, A.: *Progressive Semantic Query Answering*. In: *Proceeding of the 6th Scalable Semantic Web Knowledge Based System – International Semantic Web Conference-ISWC Workshops, Shanghai-China*, vol. vi, pp. 112–126 (2010)
17. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M., Rosati, R.: *Ontologies and Databases: The DL-Lite Approach*. In: Tessaris, S., Franconi, E., Eiter, T., Gutierrez, C., Handschuh, S., Rousset, M.-C., Schmidt, R.A. (eds.) *Reasoning Web*. LNCS, vol. 5689, pp. 255–356. Springer, Heidelberg (2009)

A Framework to Assure the Quality of Sanity Check Process

Rabia Sammi¹, Iram Masood², and Shunaila Jabeen²

¹ Air University, Islamabad, Pakistan

² University Institute of Information Technology

PAMS Arid Agriculture University

Rawalpindi, Pakistan

rabia_sammi@yahoo.com,

Iram.massod14@yahoo.com,

shunaila_jbn@yahoo.com

Abstract. Sanity check is a process which verifies that the software under development meets its basic functional requirements. In quality audits sanity check is consider as a major activity. It performs a quick test to check the main functionality of the software. Sanity check decides the completion of development phase and makes a go/no go decision to forward the software to testing phase. The depth of sanity check process varies in different scenarios and sometimes it is considered as a full quality audit. A proper and well prepared sanity check significantly reduce the time and cost of overall project. In this paper we proposed a new framework to assure the quality of sanity check process with prioritize set of activities.

Keywords: Sanity check, Sanity Testing, Software Quality Assurance.

1 Introduction

Sanity check or sanity testing is a broad and quick testing of software's basic functionality [16]. The purpose of sanity test is to verify the system's working and its basic functionality rather than finding errors in the implementation. It is a process which decides that the software development is completed and it is ready to forward to the validation phase. As validation is the last phase in SDLC so the software under development (SUD) must fulfill the basic requirements in order to be tested. Even in recursive or iterative process models, some functionality would be considered in a new release. If the software does not meets its central functionality than there is no purpose of testing the software. Sanity check performs a quick test to determine that the SUD is stable and it is reasonable to start the detailed testing. If the software fails sanity check then it is consider as being unstable [4]. Informally sanity testing means that you are requesting a check of your assumptions. If the result of the sanity check approves the assumptions then SUD is ready move forward to the rigorous and detailed testing, or otherwise, roll back the current development and start from the beginning. Sanity check is a run-time test which is used either for validating input or

the requirements. The idea of sanity checking is that whether one or more values for the functions of the software fall inside or outside of a region of validity [10].

It is reported that sanity check highly increases the quality of the software, and reduces the efforts required in validation process. It also minimizes the cost as it is performed in the early stages of development [8] [9]. Zhao [17] stated sanity check as the most potential way to reduce the cost of testing. In all software, sanity check is performed as the initial test for their working. If the quality of sanity check process is good and it is planned and performed according to the quality standards then it will result in cutting down the significant effort in terms of both time and cost. It also helps in avoiding the serious situations such as late delivery or over budget. To reduce the cost and efforts sanity check process should be effective and automated [4].

In this work we present a new framework to assure the quality of sanity check process. Quality assurance is a process or set of processes used to assess the quality of a product. Software quality assurance can be best explained in IEEE [15] as:

“A set of activities designed to evaluate the process by which the products are developed or manufactured”.

The presented framework defines a set of activities for the sanity check process. The idea of this framework is to assign priorities to each activity. The framework works iteratively for the low priority activities but if a high priority activity of the process fails then sanity test is terminated and have to start all over again.

The rest of the paper is organized as follows: Section 2 gives an overview of the domain, section 3 includes the proposed framework for sanity check process and section 4 concluded the paper.

2 Literature Review

Different companies where testing methodologies are not used are facing lots of problems like time over flow and cost over flow, but some problems are handled by sanity checking because in sanity checking faults are identified before the start of rigorous testing. Sanity check is a fast and cost effective technique, which is very useful for software development and maintenance. In previous work sanity checking is also used as regression testing [14]. The frequent use of sanity test is found in mission critical or safety critical systems because testing and retesting of these systems is frequent [13][14]. In [13] sanity test is used as a basic test in ice detection software developed for Navy ships. The software used wind speed and significant wave height as the parameters for sanity check.

Akira [14] reported that in software development organizations where sanity test is not performed, the effort of correcting the bugs increase. The software houses that do not use checking methodologies are not producing quality software products [14]. The work presented in [7] start using sanity check at early stages of requirement analysis. The requirement gathering and elicitation are the initial steps of the software development. All the documents, requirements, the modified model elements, and new generated model elements are gathered in a validation report. The review of all the generated artifacts from the validation report is used to check the quality of the team’s work and deliverables. In [7] Sanity check is used to check the existence of

functional requirements. In addition, the conformance of requirements with the standards is also checked to assure quality of the requirements. A quality requirement has to be correct, unambiguous, complete, consistent, verifiable, traceable, and modifiable [11] [12]. The identification of essential requirements works as sanity check in [11]. Because sanity check validates the existence of functional requirements. The checking of team's work against goals and expectations can also serve as a sanity check [7].

Sundmark et al. [6] uses sanity check as a way through which functional operation of component effectiveness is identified according to the component inputs and its current state during runtime. The components output based on the input of sanity check. It conforms that given input is according to the output. Sanity check allows the component interface for runtime contract checking. Due to sanity check, testing and debugging is enhanced. Sanity check is not concerned with the methods which used in debugging and methods which are using to visualize component behavior of components at runtime [6].

Some rules of software estimation are used as sanity check in terms of schedule, resource, cost, and sizing, as presented in [5]. These rules are not 100% accurate, but they can give a general idea. By implementing these rules quality of sanity check for estimation of the product is improved [5].

Sanity test is used to check the compliance of software artifacts and the software under development is presented in [3]. Whereas [2] introduced sanity check in the domain of data bases. In the context of database, traditionally data is collected and stored online after doing the syntax checking; then after going offline, data is send to the database. The errors in the data are found when some processing is done on it. The errors may be permanent or temporary. The temporary errors may occur due to transmission faults, and the permanent errors may occur due to malfunction of data collection hardware. Yorman [2] claimed that by identifying these types of errors before storing the data in the database will reduce the effort and cost involved in storage of error free data by doing sanity check. An online sanity test tool for databases called DASANEX (Data Sanity Exert) is presented in [1], which checks the data for errors This software does four tasks: checks the data syntax, apply rules to the data, modify the data rules, and make new rules after observing the data.[1]

Sanity testing reduces efforts and testing time. There are some objectives of sanity checking which are: increase testing coverage, according to the user and customer measure test coverage is the completeness of testing process; improve version stability which shows how much the product is developed; reduce quality assurance cycle time which is measured by different modules or sub systems which are used till testing completion; prediction of version quality, there are quality levels for a software products, version quality prediction states that the new version of software will be at what quality level; depends on the models and improved version reliability measured by usage profiles and number of failures [2].

The use of sanity test and its importance is shown in the above review. It is evident that sanity check proved to be beneficial in software development. But only a well managed and quality sanity test would fulfill this purpose. According to our knowledge previously no proper framework is presented to assure the quality of sanity check process.

3 Proposed Frame Work

In the proposed framework we defined a set of activities which must be carried out for a quality sanity check process. Some activities are defined as high priority activities and some as low priority activities. These activities are sanity check plan, baseline sanity check plan, configuration management of sanity check plan, traceability matrix for sanity check plan, Sanity check document, baseline sanity check document, configuration management of sanity check document, traceability matrix for sanity check document, functional requirement identification, Business process of software, change in baseline, sanity check document checking with SRS and SDS.

The activities which are assigned high priority are very important to execute, such as sanity check plan, sanity check document, and functional requirement identification. If any of these activities are not performed then the audit of the sanity check phase is considered as fail. All of these three activities have to be followed and fulfilled in order to get the audit pass report because these are the major and high priority steps to get the audit pass report. The remaining phases are of low priority and of less importance. The minimum criterion for passing the quality audit of sanity test is to perform at least four out of six low priority activities as well as all high priority activities are mandatory. That means any four out of six low priority phases have to be followed or fulfilled in order to get the complete audit pass document.

The proposed framework is shown in figure 1 and is explained as follows.

3.1 Sanity Check Phase

Sanity check is an essential test which is performed at different levels of software development. Sanity check decides whether it is rational to proceed further to test the software. It also checks that the software fulfilled its basic functionality or not. If the software would not satisfy its basic functionality then there is no need to test it further so sanity check makes the go/no go decision for rigorous testing. Here go/no go decision is to decide whether to proceed to the next stage or to remain in the current stage to enhance or correct the faults.

This phase is the starting point of our proposed framework. It is a gateway through which we can get the audit pass document successfully.

3.2 Sanity Check Plan

Sanity check plan (SCP) is a high priority phase. SCP is an important activity which includes scheduling, effort estimation, work breakdown structure and resource allocation. Planning is concerned with identifying the activities, milestones and deliverables produced so nothing can be done properly without planning. SCP is a defining document for the sanity check process. SCP must comply with the organizational standards, and evolve along with the requirements and coded system. Two indications of a useful SCP are periodic updating, understanding and acceptance by quality assurance team and quality control team.

Quality assurance team will check if the plan is according to the standards and is in the outcome of the involvement of quality assurance and quality control teams, then go decision for next step will be taken, otherwise audit is considered as fail.

3.3 Sanity Check Document

Sanity check document (SCD) is a high priority phase like SCP. SCD includes all the detailed information about the sanity check phase, such as: planning information document, effort estimation document, and revision tracking matrix. Quality assurance team will check the development of SCD, and also checks its format with the standard template for SCD approved by quality assurance department. If the SCD is developed appropriately then proceed to the next step, else fail the audit.

3.4 Functional Requirement Identification

Functional requirement identification is also a high priority phase. Sanity check is concerned with functionality of the system which is described as functional requirements, so it is necessary to identify functional requirements correctly. Furthermore, it is also necessary to check that the developed system fulfills the identified functional requirements or not.

Quality assurance team will check the functional requirements of the developed system with respect to the system requirement specification (SRS) document. If functional requirements are in accordance with SRS then move to the next phase, else the generate audit fail report.

3.5 Other Phases of Framework

These phases are of low priority in our framework. For successful audit it is necessary to fulfill any four of the low priority phases. It is the minimum criteria for a quality sanity check audit.

The low priority phases are baseline SCP & SCD, traceability matrix for SCP & SCD, configuration management of SCP & SCD, baseline management, business process of software, and SCD checking with system requirement specification (SRS) and system design specification (SDS).

Baseline SCP & SCD. The baseline is the point at which the plan and document is considered as prone to errors and freeze up to that point. Sanity Check Plan and Sanity Check Document should be baseline because all testers have to follow the plan so it should be freeze. Quality assurance team will check if the baseline is identified and is according to the standards or not.

Configuration management of SCP & SCD. Configuration management deals with how the change will be handled and integrated. It is the ability to verify that the final delivered software has all of the planned enhancements that are supposed to be included in the release, and everyone gets the updated version of the plan. Quality assurance team will check if the configuration management of the sanity check plan and sanity check document is done.

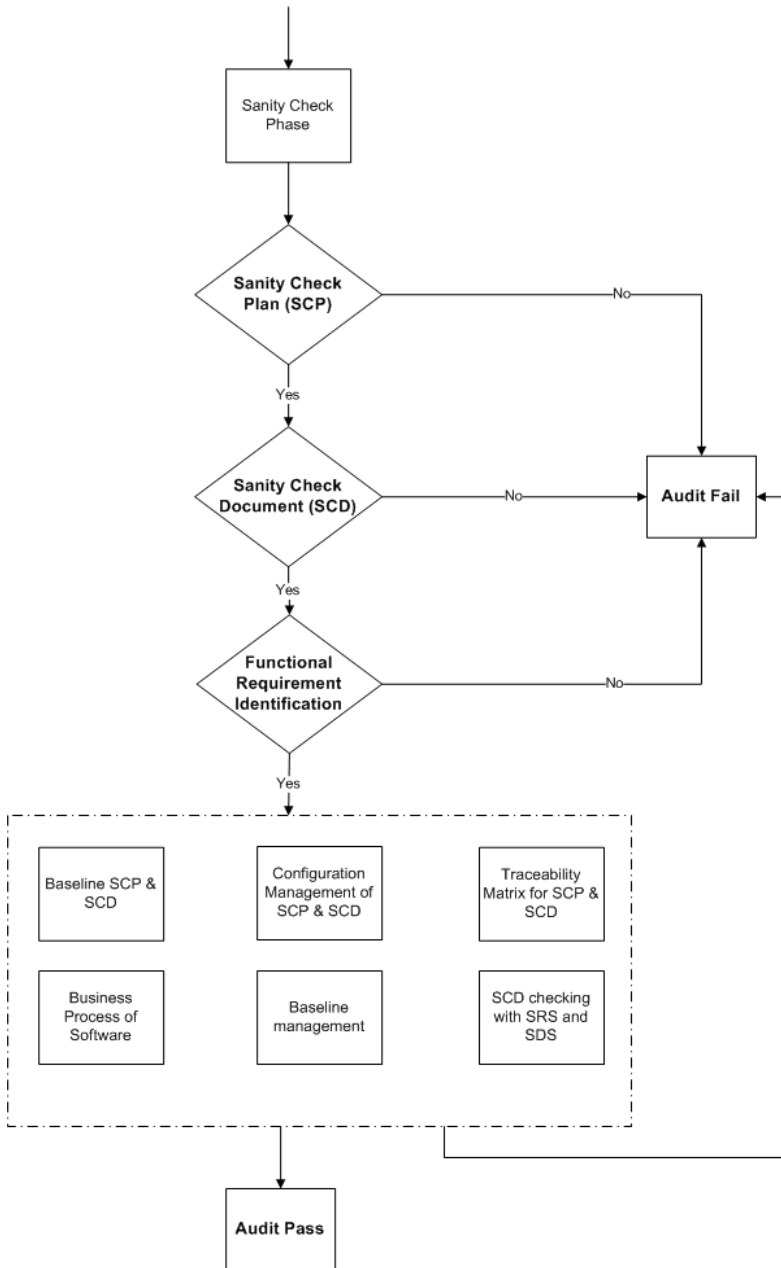


Fig. 1. Frame work to assure the quality of sanity check process

Traceability matrix for SCP & SCD. Traceability matrix is used to maintain record of different versions of the plan. When something is changed in the baseline plan, the version of new plan is stored in traceability matrix; it's easy to see what needs to be

changed in the other documents. Quality assurance team will check if the traceability matrix is established according to the standards.

Business process of software. Business process of software has significant impact on functionality of system. Some functions are dependent on others as: unless a person is authenticated one can not access the secure parts of the system. Unless the dependencies are identified, the functionality could not be correctly achieved. So the relationship of function should be identified. Quality assurance team will check if the business process is established according to the standards.

Baseline management. If a change occurred in any baseline then the change should be applied after peer review and all the stakeholders have to be notified. Quality assurance team will check if the change in baseline is established according to the standards.

Sanity check document checking with SRS and SDS. Quality assurance team will check if the sanity check document is consistent with the SRS and system design specification (SDS). SRS and SDS are important specifications which give the proper functional requirements for the developing system. From these documents quality assurance team get proper guidelines for the software functionality.

4 Conclusion

Sanity check or sanity test is a quick and broad level test of systems functionality. It decides that software fulfill its required functionality. If the sanity test fails, it is not reasonable to attempt more rigorous testing. Sanity check is proved to be an essential activity in software development. Sanity check can be done for requirements, data and software product. A well managed and quality sanity check not only reduces the time of overall project but also helpful in cutting off the cost of the project. In this work a new framework is proposed to assure the quality of sanity test. The frame work introduces the set of activities to perform sanity check. The activities are selected from the standard quality assurance practices. We prioritize the activities with respect to sanity check process in our framework.

References

1. Lahouar, S., Rahoman, S.: Design and implementation of an expert system for data sanity checking. In: Proceedings of Southeastcon 1989, Energy and Information Technologies in the Southeast, vol, April 9-12, 1989, IEEE, Los Alamitos (1989), doi:10.1109/SECON.1989.132347
2. Lev-Yehudi, Y., Perry, A.: Implementing Automatic testing is not so automatic: Lessons learned from an implementation experiment in Magic Software Enterprises (1997)
3. Filho, W.P.P.: Quality Gates in Use-Case Driven Development. In: Proceedings of the 2006 international workshop on Software quality, WoSQ 2006 (2006), ISBN:1-59593-399-9

4. Lassenius, K.R.C.: Pacing Software Product Development: A Framework and Practical Implementation Guidelines (2006), ISBN 951-22-8382-4
5. Jones, C.: By popular demand: Software estimating rules of thumb. *Computer* 29(3), 116 (1996), doi:10.1109/MC.1996.4859
6. Sundmark, D., Möller, A., Nolin, M.: Monitoring Software Components- A Novel Software Engineering Approach. In: Proceedings of the 11th Asia-Pacific Software Engineering Conference (APSEC 2004). IEEE Computer Society, Los Alamitos (2004)
7. Mead, N., Stehney, T.: Security Quality Requirements Engineering (SQUARE) Methodology (2005)
8. Li, S., Xu, J., Deng, L.: Periodic Partial Validation: Cost-effective Source Code Validation Process in Cross-platform Software Development Environment. In: Proceedings of the 10th IEEE Pacific Rim International Symposium on Dependable Computing, PRDC 2004 (2004)
9. Weller, E.F.: Lessons from three years of inspection data. *Journal of IEEE software* 10(5) (1993)
10. Trappe, W., Zhang, Y., Nath, B.: MIAMI: Methods and Infrastructure for the Assurance of Measurement Information. In: Proceedings of the 2nd International VLDB Workshop on Data Management for Sensor Networks (2005)
11. Piprani, B.: Using ORM-Based Models as a Foundation for a Data Quality Firewall in an Advanced Generation Data Warehouse. Springer, Heidelberg (2006)
12. Mellado, D., Fernández-Medina, E., Piattini, M.: A common criteria based security requirements engineering process for the development of secure information systems. Elsevier, Amsterdam (2006)
13. Coulter, R.E.: Ice Edge Detection and Icewater Classification Utilizing the ERS-1 and TOPEX Altimeters. IEEE, Los Alamitos (1994)
14. Onoma, A.K., Tsai, W.K., Poonawala, M.H., Sukanuma, H.: Regression Testing in an Industrial Environment. ACM Press, New York (1998)
15. IEEE standards, (IEEE Std 610.12-1990)
16. Fecko, M.A., Lott, C.M.: Lessons learned from automating tests for an operations support system, *Pract. Exper.* 00, 1–23 (2002)
17. Zhao, N.Y., Shum, M.W.: Technical Solution to Automate Smoke Test Using Rational Functional Tester and Virtualization Technology. 30th Annual International Conference on Computer Software and Applications Conference (COMPSAC 2006) 2, 367 (2006), doi:10.1109/COMPSAC.2006.166

Scalability of Database Bulk Insertion with Multi-threading

Boon Wee Low, Boon Yaik Ooi, and Chee Siang Wong

Faculty of Information and Communication Technology, Department of Computer Science,
Universiti Tunku Abdul Rahman, Jalan Universiti, Bandar Barat,
31900 Kampar, Perak, Malaysia.
{lowbw, ooi by, wongcs}@utar.edu.my

Abstract. The advancement of multicore processors and database technologies have enable database insertion to be implemented concurrently via multithreading programming. In this work, we evaluate the performance of using multithreading technique to perform database insertion of large data set with known size. The performance evaluation includes techniques such as using single database connection, multithreads the insertion process with respective database connections, single threaded bulk insertion and multithreaded bulk insertion. MySQL 5.2 and SQL Server 2008 were used and the experimental results show that larger datasets bulk insertion of both databases can drastically be improved with multithreading.

Keywords: Database bulk insertion, multicore processor, multi-threading and database technologies.

1 Introduction

Over the years, the steady increase of the number of cores in microprocessors has enabled parallel processing to be applied in systems such as enterprise resource planning (ERP) and customer relationship management (CRM) systems. For instance, the latest Intel Core i7 Gulftown microprocessor [1] offers up to 12 logical cores when simultaneous multi-threading is enabled. However, this increase of processing resources can only be utilized by a program if multi-threading or multi-processing techniques are used. One of such server-oriented application that may utilize multi-threading techniques is database system [3, 4]. Past research has shown that multi-threading is capable of improving the speed of database insertion. However, the scalability of such improvement with respect to various data sizes offers intriguing insight into providing overall improvement in database performance.

In this paper, we investigate the scalability of performance improvement with respect to the size of the dataset, available cores and insertion techniques such as bulk insertion. The relationships of the CPU (Central Processing Unit), the RAM (Random Access Memory), the I/O (Input/Output) transfer rate of system storages, and the performance of database bulk insertion are studied as well. The main contribution of this paper is to evaluate which insertion methods offer the best performance that suits

different database bulk insertion environment. The remaining parts of the paper are organized as follows: Section 2 discusses the related work in improving the performance of database bulk insertion by using multi-threading techniques. Section 3 details the methodology of the research, which includes experimental setup, threading methods used, and the systems utilization. The outcome of the evaluations is presented and discussed in Section 4. Finally, Section 5 concludes the paper.

2 Related Work

The conventional way of inserting data into a database is by using the sequential SQL Insert method. In order to perform data insertion in bulk, database vendors have developed specific methods so that data can be inserted at a better rate. However, current DAL (Data Access Layer) is single-threaded and no attempts of multi-threading the DAL have been made so far as the authors are able to identify.

Previously, other research has shown that multi-threading can improve the performance of database insertion. This has set the trend of utilizing thread level parallelism and performance scalability in modern software development [3].

Previous works related to parallel database systems have also been studied. Özsü and Valdúriez [19] introduced distributed and parallel Database Management System (DBMS) that enables natural growth and expansion of database on simple machines. Parallel DBMSs are one of the most realistic ways working towards meeting the performance requirements of application which demands significant throughput on the DBMS.

DeWitt and Gray [17] shows that parallel processing is a cheap and fast way to significantly gain performance in database system. Software techniques such as data partitioning, dataflow, and intra-operator parallelism are needed to be employed to have an easy migration to parallel processing. The availability of fast processors and inexpensive disk packages is an ideal platform for parallel database systems.

According to Valdúriez [18], parallel database system is the way forward into making full use of multiprocessor architectures using software-oriented solutions. This method promises high-performance, high-availability and extensibility power price compared to mainframes servers. Parallelism is the most efficient solution into supporting huge databases on a single machine.

In a research to speedup database performance, Haggander and Lundberg [16] shows that by multi-threading the database application it would increase the performance by 4.4 times than of a single threaded engine. This research was done to support a fraud detection application which requires high performance read and write processes. Therefore they found that the process would be speed up by increasing the number of simultaneous request.

Zhou et al. [15] shows that there is moderate performance increase when database is being multithreaded. He evaluated its performance, implementation complexity, and other measures and provides a guideline on how to make use of various threading method. From the experiment results, multi-threading improves the database performance by 30% to 70% over single-threaded implementation. In this research, it is also found that Naïve parallelism is the easiest to implement. However, it only gives a modest performance improvement.

In 2009, Ryan Johnson shows that by increasing the number of concurrent threads it would also increase the normalized throughput of data into a database. But there is a limit on how many concurrent threads can be used. As the number of threads used gone pass the optimal figure, it will suffer from performance deterioration due to extra overheads initiated from additional context switching as a consequence of spawning excessive number of threads. The experiment was done based on different database engines; which are Postgres, MySQL, Shore and BDB. It can be concluded that different database engine has its respective optimal number of threads. The optimal number depends on how the database was being developed. This comes to show that a detailed study on different database system is required to get the best out of each database system. The research concludes that multi-threading does help in improving database insertion speed. The paper also discovers the bottlenecks that hamper the scalability. It is overcome by introducing Shore-MT, a multi-threaded version of Shore database engine which shows excellent scalability and great performance when compared to other database engines [3].

Reference [4] uses of .NET 4 Framework to parallelize database access. From the test results, they have shown a significant increase of performance when there is a large amount of data. In contrast, there is only a slight increase of performance when the data size is small. The performance increases as much as 80.5% when it deals with a large amount of data. The experiment was done by inserting an amount of data into the database in parallel, and then retrieving the data from the database and storing it into an XML file. In this approach, multiple connections and threads access the database in parallel and all is controlled by the .NET 4 Framework.

From all previous research, we can see great potential in multi-threading database systems. It is proven that by parallelizing the system, it would have a moderate to significant gain in performance at a lower cost. Therefore with the right threading method and insertion method, we are able to improve database performance. This proves the potential in multi-threading database systems.

3 Methodology

This paper focuses on two database systems, Microsoft SQL Server 2008 Enterprise and MY SQL 5.1 by Oracle. All the evaluations done in this work were conducted on a same machine. We observe the performance of various insertion methods with multithreading implementation.

3.1 Environment Settings of the Experiment

The machine used is comprised of an Intel Core 2 Quad Q9400 2.66 GHz with 3.93 GB of RAM with Windows XP Professional Service Pack 3. The hard-disk used in this test bed is a 320GB Seagate Barracuda with rotational speed of 7200rpm (revolution per minute), and is capable of performing 78 MB/s data transfer rate [5]. At start the machine consumes 438 MB of RAM and 0% CPU utilization.

Test program was being developed and tested on Visual Studio 2010 with .NET 4.0 Framework using C#. MY SQL Connector .NET 6.2.4 adapter is being used to execute the InsertLoader for MySQL database.

The test data consists of strings with 302 random characters each. It comprises alphabets, numeric and symbols. These strings are stored in flat file format and the file size ranges from 1 to 80,000 rows. The same set of files is used for the entire experiment.

The database consists of one table with 2 columns. The first column is an auto-increment numeric counter which is set to integer and the second column is to store the rows from the flat file which is set to VARCHAR(MAX). This is applied to both databases that are evaluated in this paper.

3.2 The Overview of the Experiment Process

The reading process is done by using a single thread. Then it will distribute the rows into multiple files depending on the number of threads is going to be created. The reader will write the file into either flat file format or XML format depending on the database engine.

Table 1. Evaluation on various insertion methods and the number of threads used respectively

Test Number	Insertion Method	Number of Threads
1	Sequential SQL Insertion (SQL Server 2008 & MySQL 5.2)	1
		2
		3
		4
2	Import Loader	1
		2
		3
		4
3	Bulk Copy	1
		2
		3
		4

3.2.1 Threading Method

Throughout the experiments, threads are manually spawned in order to maintain a controlled environment. The codes below show how the program is being threaded where two threads are used.

```
//create and start threads
ThreadStart threadDelOne = new ThreadStart
(insOne.RunInsertion);
ThreadStart threadDelTwo = new ThreadStart
(insTwo.RunInsertion);

Thread threadOne = new Thread(threadDelOne);
Thread threadTwo = new Thread(threadDelTwo);
threadOne.Start();
threadTwo.Start();
```

```
//thread join
threadOne.Join();
threadTwo.Join();
```

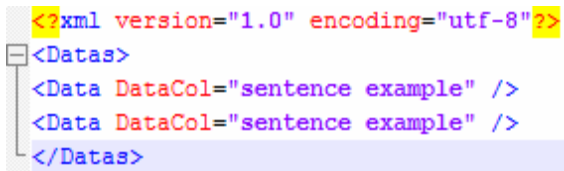
3.2.2 Sequential Insertion

Sequential insertion is done by using the standard SQL insert command and each command would insert one row. Transaction is being used in this process where the whole block will be committed after the last data is inserted. Rollback is being used if there is an error [6]. The same code is being used for 1, 2, 4 and 8 threads. Before inserting the row, it is being formatted into compatible SQL command by replacing certain characters to work with SQL command formatting. For sequential, the test is done with and without transaction.

```
for (int i = 0; i < dataList.Count; i++) {
    sqlStr = "INSERT INTO TestTbl(DataCol)
VALUES(N" + dataList[i] + ")";
    sqlCmd = new SqlCommand(sqlStr, conn,
transaction);
    sqlCmd.ExecuteNonQuery();
}
transaction.Commit();
```

3.2.3 SQLBulkCopy Insertion

SQLBulkCopy is a .NET4 function to insert data in bulks into SQL Server 2008 [8]. It receives XML's and inserts them. The format for the XML file is as shown in Fig. 1. This method is tested by using 1, 2, 4, and 8 threads; where the number of individual XML's are read according to the number of threads used respectively. For example, if 8 threads are used, they will read from 8 individual XML's.



```
<?xml version="1.0" encoding="utf-8"?>
<Datas>
  <Data DataCol="sentence example" />
  <Data DataCol="sentence example" />
</Datas>
```

Fig. 1. An example of XML formatting

The codes below show how the SQLBulkCopy insertion is being done.

```
dataSet dataSet = new DataSet();
dataSet.ReadXml(xmlFileName);
sourceData = dataSet.Tables[0];

//perform insertion
using (SqlConnection conn = new SqlConnection(connStr))
```

```

{
    conn.Open();
    using (SqlBulkCopy bulkCopy = new
        SqlBulkCopy(conn.ConnectionString))
    {
        bulkCopy.ColumnMappings.Add("DataCol", "DataCol");
        bulkCopy.DestinationTableName = "TestTbl";
        bulkCopy.WriteToServer(sourceData);
    }
conn.Close();
}

```

3.2.4 MySQL Bulk Loader Insertion

MySQL Bulk Loader from the MySQL .NET Connector 6.2.4 [7] is used for this experiment. It receives flat files and inserts them using the MySQL import loader. The number of files created would depend on the number of threads used. This method is being tested with 1, 2, 4, and 8 threads. The following codes illustrate how the import loader is performed.

```

//perform import loader
try    {
    MySqlBulkLoader myBulk = new MySqlBulkLoader(conn);
    myBulk.Timeout = 600;
    myBulk.TableName = "testDatabase.testTbl";
    myBulk.Local = true;
    myBulk.FileName = fileName;
    myBulk.FieldTerminator = "";
    myBulk.Load();
}

```

3.2.5 System Utilization

In the next experiment, the RAM, CPU and hard disk drive utilization are captured throughout the insertion period. This is done during 70,000 to 80,000 rows on all the insertion methods, number of threads and database engines as shown in Table 1. A sample is captured every 30 seconds and the average from the samples would be taken as the result [9]. Codes below illustrates the system utilization is being captured.

```

PerformanceCounter cpuUsage = new
PerformanceCounter("Processor", "% Processor Time",
    "_Total", true);

PerformanceCounter memoryAvailable = new
PerformanceCounter("Memory", "Available MBytes");

PerformanceCounter physicalDiskTransfer = new
PerformanceCounter("PhysicalDisk", "Disk Bytes/sec",
    "_Total", true);

startMemory = totalMemoryCapacity -
memoryAvailable.NextValue();

```

4 Results and Discussion

The test data ranges from 1 to 80,000 rows and we capture the elapsed time taken to insert the data. The same test data are being used on both database engines with the method discussed in chapter 3.

4.1 SQL Server 2008

Multi-threading has a significant improvement. At 50,000 rows, performance increase as much as 67% using multithreaded insertion method. Code with transaction increases the performance by 24%. Therefore multi-threading and transaction proves to improve the database insertion performance. Fig. 2 shows the performance increase between 1 and 8 threads using sequential insertion. But when the data size is small, the overhead of spawning the threads is too costly and the performance would deteriorate. In contrast with that, it implies that it is best not to multithread the database insertion when the data size is small. This is reflected in Fig. 3.

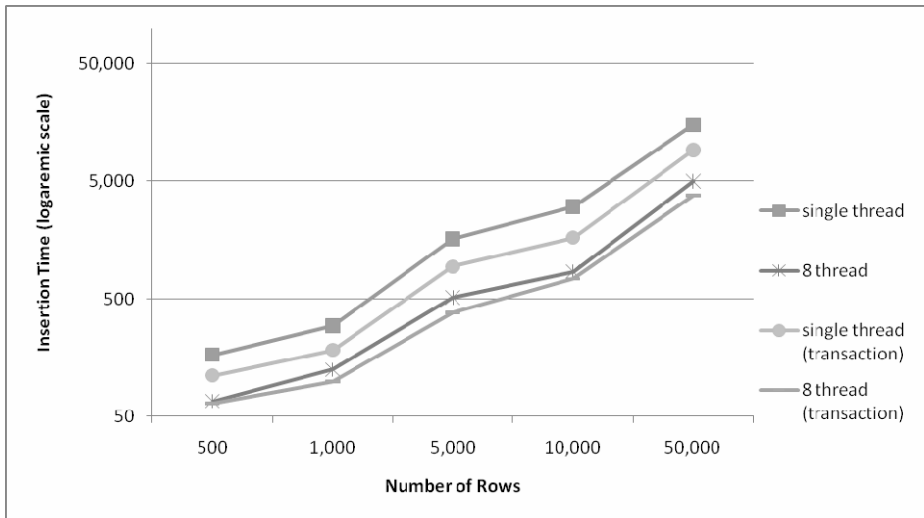


Fig. 2. Sequential insertion with and without transaction used

From Fig. 3, it showed that single threaded bulk copy can outperform the multithreaded insertion with transaction enabled. As the data grow larger, bulk copy is becoming more efficient.

However, the performance of bulk copy can be further improved by using multiple threads. Fig. 4 shows the performance of BulkCopy compared to sequential insertion with a large data size. At 80,000 rows, the performance increase by 43% when threaded with eight threads.

From the experiment, we observed that the performance of the insertion methods is dependent on the data size. The following table is the detail of the observation.

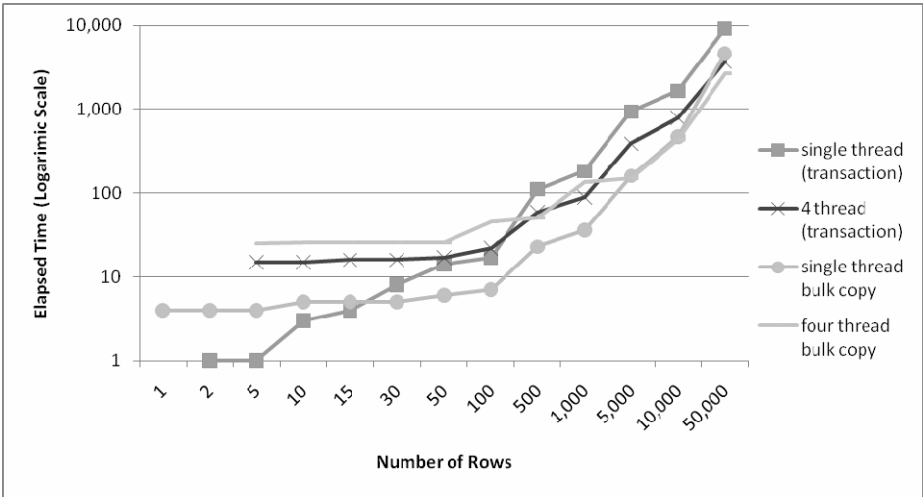


Fig. 3. Comparison between BulkCopy and sequential insertion with transaction

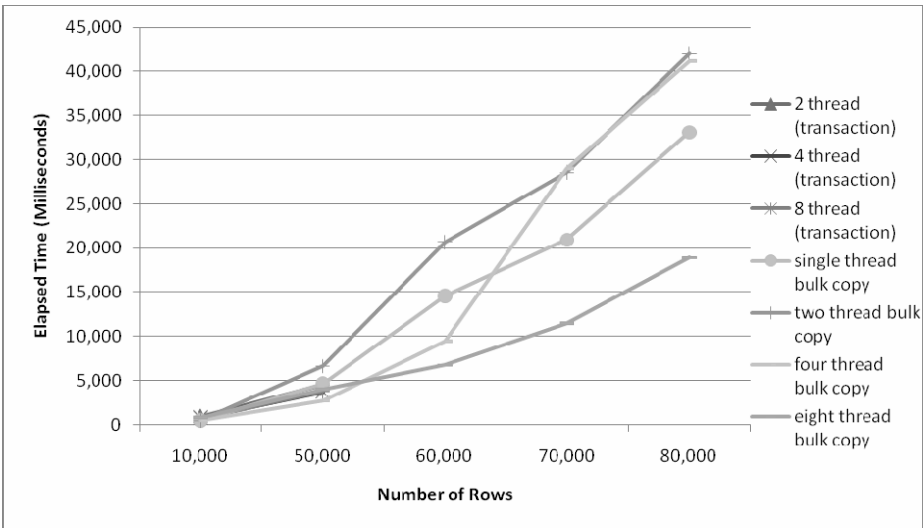


Fig. 4. Comparison between BulkCopy and sequential insertion

Table 2. SQL Server 2008 Insertion method for specific data size range

Number of Rows	Threading Method
1 to 29	Single threaded sequential insertion with transaction
30 to 5,000	Single threaded BulkCopy
5,001 to 50,000	Four threads BulkCopy
50,000 to 80,000	Eight thread BulkCopy

4.2 MySQL 5.2

On the other hand, MySQL does not have significant performance improvement when being threaded. MySQL boost sequential insertion performance by 99.5% when transaction code is being used. To insert 50,000 rows without transaction, it requires approximately 22 minutes compared to 6.3 seconds with transaction. Fig. 5 shows the sequential insertion performance.

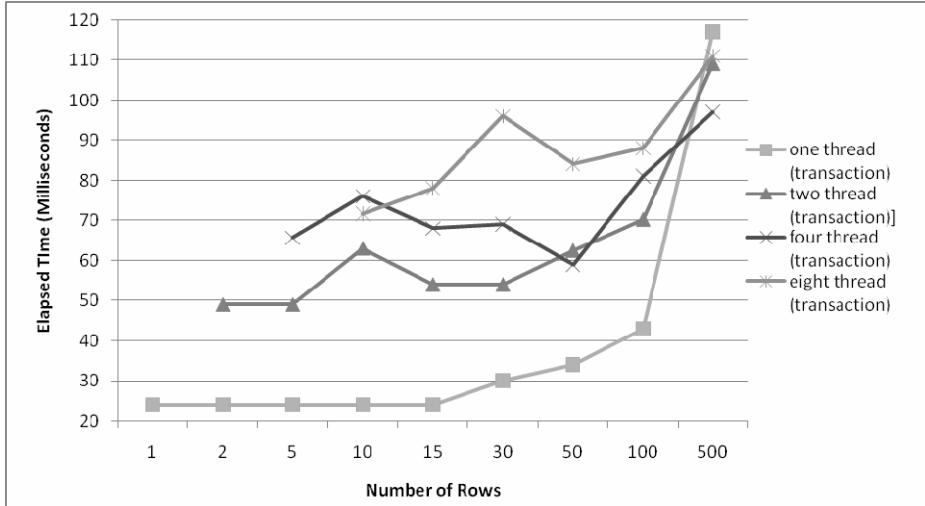


Fig. 5. Comparison between different numbers of threads using sequential insertion

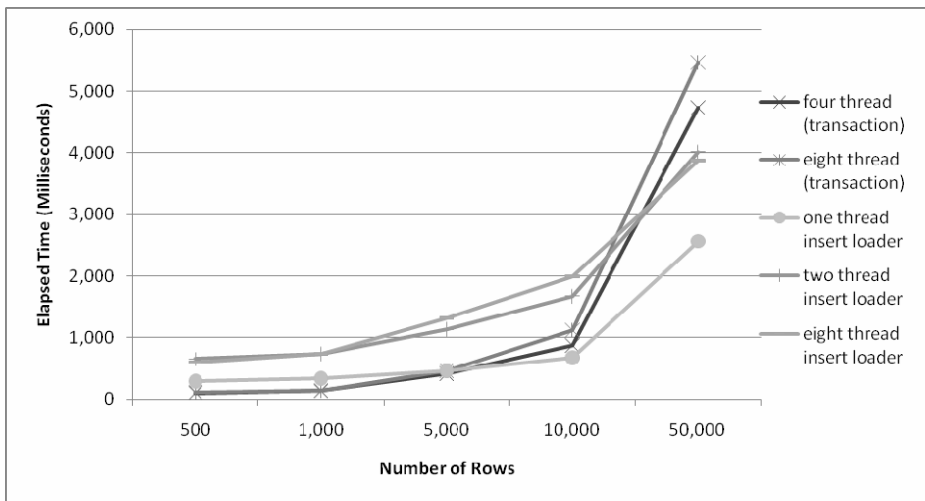


Fig. 6. Comparison between sequential insertion and insert loader

From Fig. 6, the multi-threading works well when data size is in the range of 501 to 5,000, in these range four threads improve the insertion performance by 42.5%. However, the performance plunges by 49% when it is being spawned with eight threads compared to single threaded for data size with 100 rows. Even with insert loader, single threaded still performs best. When insert loader is being spawned with eight threads, insertion performance plummet by 65.4% compared to single threaded at 80,000 rows. Insert loader performs best when the data size is large and single threaded.

The following is the observation made by from our experiment using MySQL 5.2. It is similar to MS SQL Server 2008 that the performance of the insertion methods is dependent on the data size. However, the insertion method varies.

Table 3. MySQL 5.2 Insertion method for specific data size range

Number of Rows	Threading Method
1 to 500	Single threaded sequential insertion with transaction
501 to 5,000	Four Threads sequential insertion with transaction
5,001 and 50000	Single threaded insert loader

4.3 System Utilization

Besides that, we observe the system utilization when the insertion process is executed. Observations are made only with samples sizes ranging from 70,000 to 80,000 rows as they show the most significant system utilization.

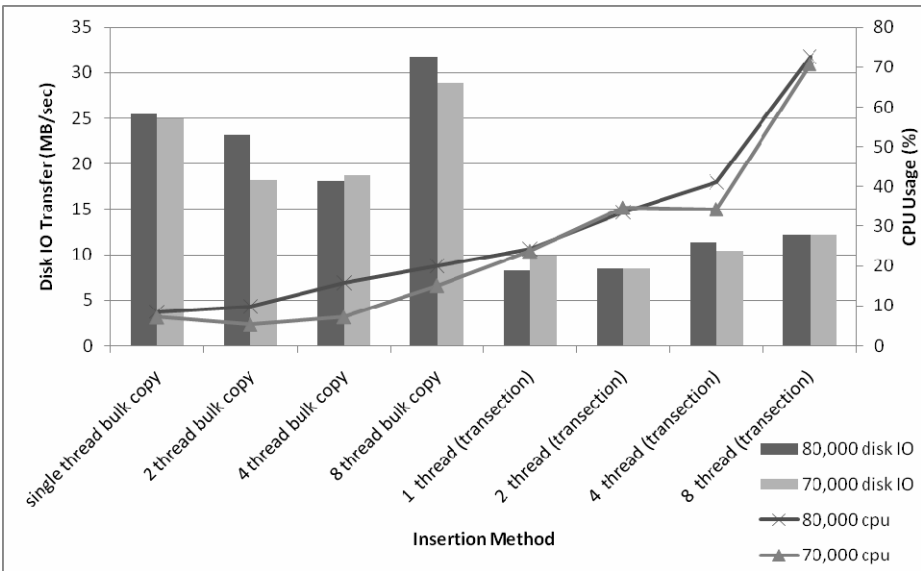


Fig. 7. System utilization between Disk I/O and CPU usage for SQL Server 2008

Fig. 7 shows the system utilization of SQL Server 2008, we found that as the number of threads increase the overall machine utilization increase as well. In general, bulk copy has high IO traces while normal SQL insertion with transaction has high CPU traces relatively. Similar observation was seen with MySQL 5.2 in the following figure.

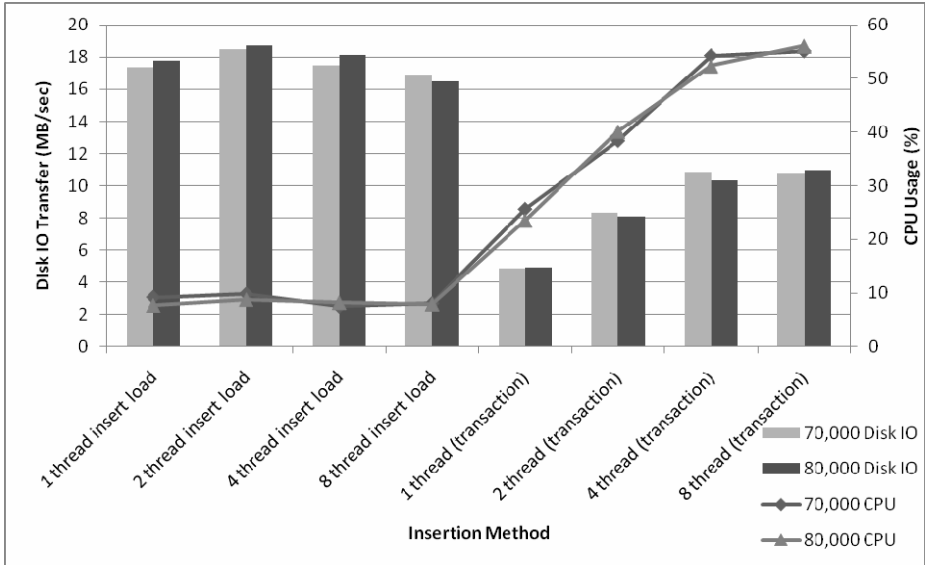


Fig. 8. System utilization between Disk I/O and CPU usage for MySQL 5.2

5 Conclusion

Although, the advancement of multicore processors is encouraging multithreaded application to be developed, we found that the performance of the insertion function of a database does not necessary improve proportionately with the number of threads used. Multithreading did improve the performance of both of the databases' insertion function but the speed up is very dependent on the underlying architecture of the database system. Therefore, this work suggests that software developers should investigate the performance of multithreaded operations on databases before designing any system.

References

1. Intel® Core™ i7 Processor Extreme Edition,
<http://www.intel.com/products/processor/corei7EE/index.html>
2. Intel® Core™ i7 – 920 Desktop Processor Series Product Specifications,
<http://ark.intel.com/Product.aspx?id=37147>

3. Johnson, R., Ippokratis, P., Hardavellas, N., Ailamaki, A., Falsafi, B.: Shore-MT: A Scalable Storage Manager for the Multicore Era. In: Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, pp. 24–35. ACM, New York (2009)
4. Verenker, A.: Using.NET4 Parallel Programming Model to Achieve Data Parallelism in Multi-tier Applications, MSIT, Microsoft Corporation (2010)
5. Seagate Barracude 7200.10 SATA 3.0Gb/s 320-GB Hard Drive,
<http://www.seagate.com/ww/v/index.jsp?vgnextoid=2d1099f4fa74c010VgnVCM100000dd04090aRCRD>
6. TransactionScope Class,
<http://msdn.microsoft.com/en-us/library/system.transactions.transactionscope.aspx>
7. Using the Bulk Loader,
<http://dev.mysql.com/doc/refman/5.1/en/connector-net-programming-bulk-loader.html>
8. SqlBulkCopy Class,
<http://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlbulkcopy.aspx>
9. Performance Counter Constructor, <http://msdn.microsoft.com/en-us/library/xx7e9t8e.aspx>
10. Bunn, J.J., Holtman, K., Newman, H.B.: Object Database Scalability for Scientific Workloads. Technical report, California Institute of Technology (2000)
11. Thread Class,
<http://msdn.microsoft.com/en-us/library/system.threading.thread.aspx>
12. Lui, D., Wang, S.: Analysis of Database Workloads on Modern Processors. In: Proceedings of the 1st SIGMOD PhD Workshop on Innovation Database Research 2007, pp. 63–68. ACM, New York (2007)
13. Performance Monitoring,
<http://www.csharp4help.com/2006/05/performance-monitoring/>
14. How to Create and Terminate Thread (C# Programming Guide),
<http://msdn.microsoft.com/en-US/library/7a2f3ay4v=VS.80.spx>
15. Zhou, J., Cieslewicz, J., Ross, K.A., Shah, M.: Improving Database Performance on Simultaneous Multithreading Processors. In: Proceedings of the 31st International Conference on Very Large Data Bases 2005, pp. 49–60. VLDB Endowment, Norway (2005)
16. Haggander, D., Lundberg, L.: Multiprocessor Performance Evaluation of a Telecommunication Fraud Detection Application. In: ARTES Graduate Student Conference, Sweden (1999)
17. DeWitt, D., Gray, J.: Parallel Database Systems: The Future of High Performance Database Processing. Commun. ACM 35, 85–98 (1992)
18. Valduriez, P.: Parallel Database Systems: Open Problems and New Issues. J. Distributed and Parallel Databases 1, 137–165 (1993)
19. Özsü, M.T., Valduriez, P.: Distributed and Parallel Database System. J. ACM Computing Surveys 28, 125–128 (1991)

Towards Unit Testing of User Interface Code for Android Mobile Applications

Ben Sadeh, Kjetil Ørbekk, Magnus M. Eide, Njaal C.A. Gjerde, Trygve A. Tønnesland, and Sundar Gopalakrishnan

Department of Computer and Information Science,
Norwegian University of Science and Technology,
Trondheim, Norway

{sadeh,orbekk,magnuei,njaalchr,
tonnesla}@stud.ntnu.no,
sundar@idi.ntnu.no

Abstract. As the availability and popularity of mobile applications grows, there is also an increased interest for them to be solid and well tested. Consequently, there is also an interest in assessing the correctness of their system rapidly, since smart phone applications usually develop quickly and have a lower lifecycle as compared to desktop applications. We are specifically interested in an efficient way of testing the Graphical User Interface (GUI), as it is both central to the user experience and harder to evaluate than standard business logic. This research paper is a study on the different ways to assess the validity of the GUI code for an Android mobile application with special focus on unit testing. It describes the available testing techniques and details the difficulty in writing unit tests for GUI code. Finally, the study gives a recommendation based on the different testing approaches available, followed by a discussion of both the implications and limitations of the findings.

Keywords: Unit testing, Integration testing, GUI, Android application, Robolectric, Model-View-Controller (MVC), Model-View-ViewModel (MVVM), Test-driven development (TDD).

1 Introduction

The smart phone market is seeing a stable growth, with more and more people depending upon their mobile devices to manage their email, entertainment, shopping and scheduling. However, quick development processes of mobile applications may come at the cost of quality assurance [1].

This study proposes a testing approach for Android mobile applications that recognizes and fits the smartphone's fast-paced development cycle by focusing on the GUI and unit testing [2].

However, unit testing the GUI is difficult [3]. Consequently, several methods have been devised in order to test classes with dependencies in a more practical way [4]. Additionally, as modern GUIs are continuously evolving in complexity, it becomes harder to establish which parts are relevant to testing [5]. Nevertheless, testing the GUI is important for an application's resilience and chance of success [6,7].

This paper explores the different methods of assessing the GUI in an Android Activity [8] in special relation to unit testing. Section 2 states our motivation and goals for the research and briefly presents some alternate methods for GUI testing an Android activity. Section 3 outlines the steps taken to successfully unit test an Android activity. Then, Section 4 compares the different methods of unit testing to determine which one fits the research goals. Finally, Section 5 concludes the paper with possible future research.

2 Background and Related Work

In this research paper we are interested in unit testing the GUI code of an Android mobile application. Since the testing process is difficult to handle and important for the user experience, this paper has been written with the following research questions in mind:

- RQ1. What are the different methods of assessing the GUI code in an Android activity?
- RQ2. Is unit testing of the GUI code on the Android platform feasible?
- RQ3. If so, is unit testing the GUI code on the Android platform beneficial, or does instrumentation testing suffice?

2.1 Testing Methodologies

Android Instrumentation test. Currently, testing the GUI in applications is based on structuring the code in such a way that as much logic as possible is separated from the interface code. Given this distinction, the GUI can be tested using standard instrumentation tests, which are included in the Android Software development kit (SDK).

In Android's own Instrumentation Testing Framework [9], the framework launches an emulator and runs the application and its test simultaneously, allowing the testing to interact with the whole application. Consequently, these instrumentation tests can be classified as integration tests.

Since this method requires the tests to be ran inside an emulator, it performs slower while being more difficult to isolate.

Model-View-ViewModel. The MVVM pattern [10,11] uses data bindings to separate the GUI code from the GUI components. This software architecture fits our research goals for unit testing GUI code, but is currently not supported on the Android platform.

3 Suggested Testing Approach

An essential part of GUI code is to interact with the graphical components on the screen, such as buttons and text fields. A well-designed MVC application separates the GUI code from the business logic. In this case, the controller's job is to receive interactions from the user, such as a button click, and react to the interaction, perhaps involving requests to the business logic.

Unit testing a controller in such an application is challenging, but possible with commonly used techniques for unit testing business logic [12]. This section will take advantage of mentioned techniques using a simple example program containing a method in the controller class to be tested. The approach involves breaking the dependencies to the user interface framework, and optionally to the business logic. In Subsection 3.1 the example application and the method to be tested are described. Then, Subsection 3.2 covers the steps taken to unit test the method using the standard Eclipse environment. Finally, Subsection 3.3 outlines the convenience of unit testing using an assisting framework.

3.1 Example Application

The example program will be a custom made calculator. It supports addition and subtraction of numbers, and has a user interface similar to traditional pocket calculators, as illustrated in Figure 1.



Fig. 1. The calculator application with the add and subtract functions

The calculator contains three main classes that are illustrated in Figure 2.

CalculatorButton. This is an enumerator with one value for each button of the calculator. It maps an Android component ID to the CalculatorButton value that is used to represent said button. For example, the '+' button in the user interface maps to CalculatorButton.B_add.

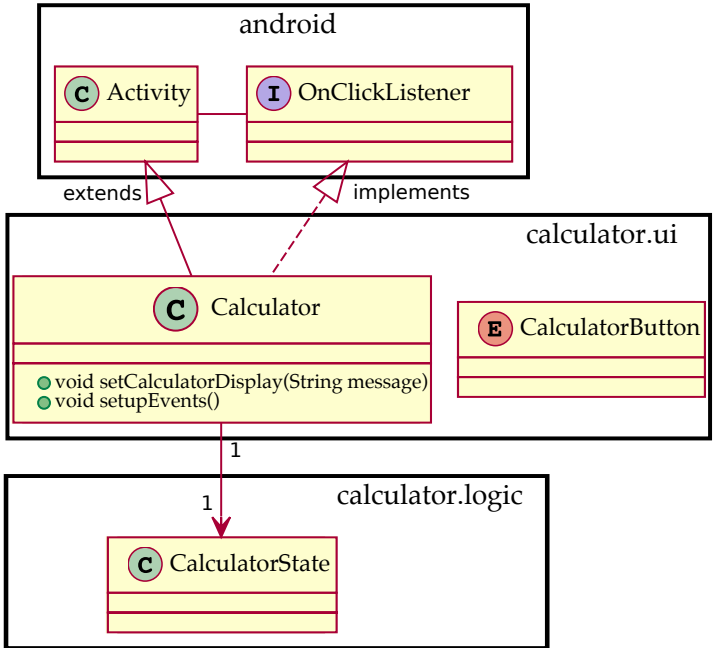


Fig. 2. The main classes in the calculator application before testing

Listing 1.1. Original `onClick()` implementation

```

public void onClick(View view) {
    // Get the token that 'view' maps to
    CalculatorButton button = CalculatorButton.findById(view.getId());
    calculatorState.pushToken(button);
    updateDisplay();
}

```

CalculatorState. The business logic is handled by this class. It accepts a `CalculatorButton` as its input and handles the state of the calculator when numbers are received.

Calculator. This class is the Android Activity of the application. It handles the user interaction by listening to click events in the user interface, done by the `onClick()` method as shown in Listing 1.1.

`onClick()` performs a two-way communication with the user interface: It retrieves the button clicked and updates the display, and to do this correctly, it needs to interact with the business logic. `updateDisplay()` is a simple method that was tested using the same techniques as `onClick()`.

3.2 Standard Environment Approach

In this approach, the default Eclipse environment is considered for the Android development [13]. However, out of the box it doesn't permit access to any of the Android classes, and so it is not possible to initialize the GUI classes such as the `Calculator` class.

Avoiding Initializing Classes. By extracting the `onClick()` method into a different class, say `CalculatorClickListener`, the code can be tested without initializing `Calculator`. If `CalculatorClickListener` implements the `OnClickListener` interface, it can act as the click listener for `Calculator`, but this prevents `CalculatorClickListener` from being instantiated. Consequently, the proposed approach works around the issue by creating a class that inherits from the class that implements `onClick()`, as shown in Listing 1.2.

The proposed approach instantiates `RealCalculatorClickListener` in the unit test. `CalculatorClickListener` is not supposed to contain any code, and therefore it should not require testing. However, in this implementation, `RealCalculatorClickListener` takes arguments in its constructor, meaning that `CalculatorClickListener` must have a constructor as well.

Since Android classes cannot be instantiated in this environment, any classes extending or implementing them cannot be tested. Therefore, the constructor of `CalculatorClickListener` remains untested.

Interacting with Android Components. Code that interacts directly with Android classes, such as `onClick()`, cannot run in a unit test because they cannot

Listing 1.2. CalculatorClickListener

```

class RealCalculatorClickListener {
    public void onClick(View view) {
        // Definition omitted
    }
}

class CalculatorClickListener extends RealCalculatorClickListener
    implements OnClickListener {
    // Empty class
}

```

Listing 1.3. ViewIdGetter

```

class ViewIdGetter {
    int getId(View view) { return view.getId(); },
}

class RealCalculatorClickListener {
    private ViewIdGetter viewIdGetter;
    RealCalculatorClickListener(ViewIdGetter viewIdGetter) {
        this.viewIdGetter = viewIdGetter;
    }

    public void onClick(View view) {
        int viewId = viewIdGetter.getId(view);
        // Remainder of definition omitted
    }
}

```

be instantiated. The solution in the standard environment is to extract the code that performs the interaction into a separate class, which then can be faked in the unit test, as illustrated in Listing 1.3

This leaves `ViewIdGetter.getId()` untested because it requires a `View` instance, and by extracting similar statements, one is able to minimize and isolate the untested code. Figure 3 provides an overview of the calculator classes after the refactoring. `onClick()` can now be unit tested using fake objects, as shown in Listing 1.4

3.3 Robolectric Approach

In order to unit test `Calculator` in the standard environment, we had to refactor the code and avoid initializing the Android framework classes. Alternatively, we could provide our own replacement definitions for the Android classes that would be possible to initialize. Fortunately, this effort has already been made by Pivotal Labs in an open framework called Robolectric [14].

Robolectric provides lightweight mock versions of the Android framework classes, called shadow objects. Moreover, they can be initialized and used in conjunction with unit tests. For example, setting the contents of a Robolectric `TextView` allows for retrieving the same contents during a test.

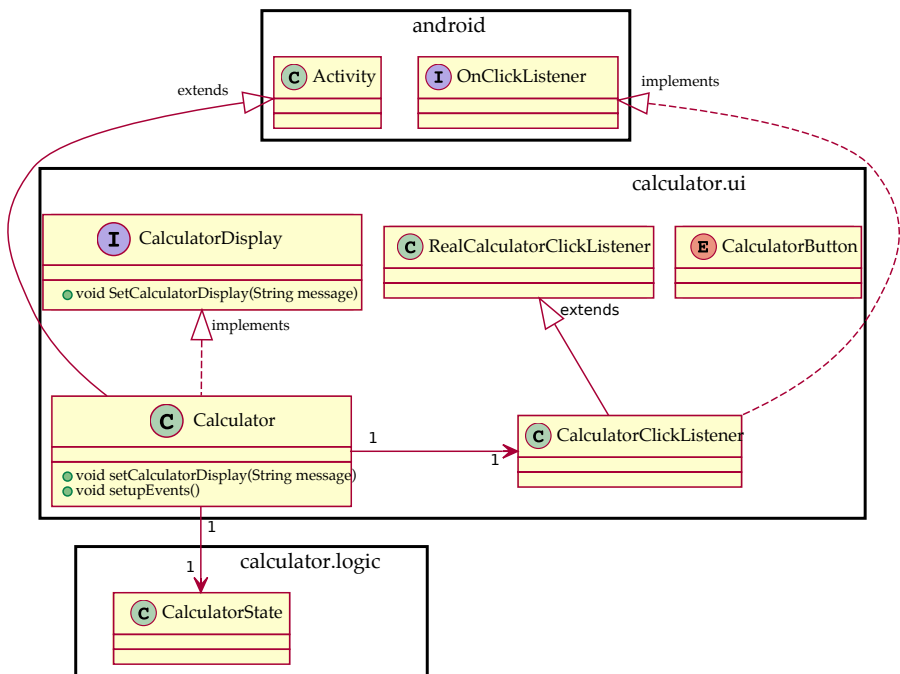


Fig. 3. The main classes in the calculator application after testing

Listing 1.4. Testing the CalculatorClickListener

```

public class CalculatorClickListenerTest {

    static class FakeCalculatorDisplay implements CalculatorDisplay {
        public String display;
        public void setCalculatorDisplay(String message) {
            display = message;
        }
    }

    static class FakeViewIdGetter extends ViewIdGetter {
        public static final CalculatorButton CLICKED_BUTTON =
            CalculatorButton.B_05;
        int getId(View unused) { return CLICKED_BUTTON.getId(); }
    }

    static class FakeCalculatorState extends CalculatorState {
        public CalculatorButton receivedToken;
        public static final String DISPLAY = "Display:FakeCalculatorState";

        public void pushToken(CalculatorButton button) {
            assertEquals(null, receivedToken);
            receivedToken = button;
        }

        public String getDisplay() { return DISPLAY; }
    }

    private RealCalculatorClickListener calculatorClickListener;
    private FakeCalculatorState calculatorState;
    private FakeCalculatorDisplay calculatorDisplay;
    private FakeViewIdGetter viewIdGetter;

    @Before
    public void setUp() {
        calculatorState = new FakeCalculatorState();
        calculatorDisplay = new FakeCalculatorDisplay();
        viewIdGetter = new FakeViewIdGetter();
        calculatorClickListener = new RealCalculatorClickListener(
            calculatorState, calculatorDisplay, viewIdGetter);
    }

    @Test
    public void testOnClick() {
        calculatorClickListener.onClick(null);
        assertEquals(FakeViewIdGetter.CLICKED_BUTTON,
            calculatorState.receivedToken);
        assertEquals(FakeCalculatorState.DISPLAY,
            calculatorDisplay.display);
    }
}

```

Listing 1.5. Testing Calculator using the Robolectric framework

```

public class CalculatorTest {

    @Test public void testOnClick() {
        Calculator calculator = new Calculator();
        calculator.onCreate(null);

        View fakeView = new View(null) {
            @Override public int getId() {
                return CalculatorButton.B_04.getId();
            }
        };

        calculator.onClick(fakeView);
        TextView display = (TextView)calculator.findViewById(
            R.id.CalculatorDisplay);
        assertEquals("4.0_", display.getText());
    }
}

```

Consequently, by using the Robolectric framework, the Calculator class can be tested with no refactoring, as illustrated in Listing [1.5](#).

4 Results and Discussion

The Calculator application was successfully unit tested in the standard environment, but only after a significant amount of refactoring and boilerplate code. Therefore, this approach may become unmanageable for larger applications.

However, the Robolectric framework makes it easy to write unit tests by requiring fewer extra steps and abstractions.

This study aims for efficiency in unit testing the GUI code in an Android mobile application. By making use of the Robolectric framework, certain qualities that are important to this research can be achieved. This paper aspires for and achieves:

- tests that run fast
- tests that are relevant
- code that is easy to maintain

Based on the initial research questions and list above, there are several categories of software tests that are of interest.

4.1 Automated Software Testing Categories

a) Unit Testing

To ensure that the individual components in a program are working, we need to assess that the smallest building blocks are built correctly. As a result, Unit tests [\[15,16\]](#) are run on individual functions and some times even whole classes in

isolation from the rest of the application. Thus, design guidelines and techniques for breaking dependencies have been developed. For example, combination of the Dependency Injection design pattern [17] and fake objects can be used to allow unit testing of a class with dependencies that would otherwise make it hard to test.

Similarly, unit testing a GUI is similar to testing a class with several external dependencies, because the interaction with a GUI framework represents a black box to the unit test.

Because unit tests cover specific parts of the program, they offer the advantage of running quickly and independent of the rest of the application.

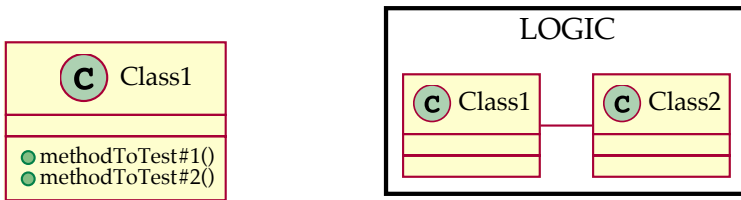
b) Integration Testing, Limitations

After the different components have been tested, they can be put together to see whether they perform as expected.

Integration testing [18] is performed by combining multiple parts of the application and is useful for checking that the different parts of the program are working together.

Integration testing is often relevant since it covers larger parts of the program. However, these tests run slower due to the added dependencies, especially when ran inside an emulator.

Figure 4 illustrates the difference between a unit test and an integration test.



(a) Unit Testing: One isolated component is tested

(b) Integration Testing: Interaction between two or more components is tested

Fig. 4. Illustration of Unit and Integration testing

4.2 Results

The `onClick()` method was tested¹ using three different methods, summarized in Table 1. Furthermore, comparison of the methods in relation to the research goals is illustrated in Table 2.

¹ Computer specifications: Intel Core 2 Duo E7500, 4 GB RAM, Debian GNU/Linux, Eclipse Helios

Table 1. Summarization of test approaches for the Calculator application

Method	Type of test	Test runtime
Android Instrumentation	Integration test	5.629 sec
Standard environment	Unit test	0.69 sec
Robolectric	Unit test	1.16 sec

Table 2. Comparison between the selected methods

Factors	Android Instrumentation (Integration test)	Standard environment (Unit test)	Robolectric (Unit test)
Ease of writing tests	++	-	+
Ease of maintenance	+	--	+
Error localization	--	-	++
Relevance	+	+	+
Speed	--	++	+

The notation is explained in the following table:

++	stands for	very good
+	"	good
-	"	unsatisfactory
--	"	very unsatisfactory

For more complex applications, using the Robolectric framework is likely to be more practical, because it allows developers to unit test their classes in isolation without having to maintain a collection of fake objects and interfaces.

Because of its nature, standard Unit testing will remain as the quickest testing method. However, classes that are refactored to allow for unit testing makes them difficult to maintain correctly. On the other hand, Integration tests are well supported and simple to run, but lack the speed and error localization that unit tests have. By using the Robolectric framework, one can achieve the speed of unit tests together with the ease of writing found in the integration tests.

However, the Robolectric approach is not a complete replacement for instrumentation tests as it does not test the actual graphical components. Moreover, this recommendation depends on the Robolectric framework to be written correctly, as it assumes responsibility for returning the accurate assessment.

5 Conclusion and Future Work

This paper explores the different options developers have for assessing the correctness of their Android mobile application.

A GUI component was successfully unit tested by adding extra code and abstractions.

Robolectric allowed tests to be written to said component with less refactoring of the original source code, and the resulting tests were fast and provided relevant test coverage of the GUI code. For this reason, unit testing GUI code is likely to benefit Android developers.

Our research currently only applies to our example application, and in future studies, we wish to expand test coverage to larger programs to obtain additional confidence in recommending unit testing with Robolectric for more complex applications and systems.

References

1. Zhifang, L., Bin, L., Xiaopeng, G.: Test automation on mobile device. In: Proceedings of the 5th Workshop on Automation of Software Test, AST 2010, pp. 1–7. ACM, New York (2010)
2. Hwang, S.M., Chae, H.C.: Design & implementation of mobile GUI testing tool. In: Proceedings of the 2008 International Conference on Convergence and Hybrid Information Technology. IEEE Computer Society Press, Los Alamitos (2008)
3. Hamill, P.: Unit Tests Framework. O'Reilly, Sebastopol (2004)
4. Brooks, P., Robinson, B., Memon, A.M.: An initial characterization of industrial graphical user interface systems. In: ICST 2009: Proceedings of the 2nd IEEE International Conference on Software Testing, Verification and Validation (2009)
5. Cai, K.Y., Zhao, L., Hu, H., Jiang, C.H.: On the test case definition for GUI testing. In: Fifth International Conference on Quality Software, QSIC 2005 (September 2005)
6. Memon, A.M.: A comprehensive framework for testing graphical user interfaces. Ph.D (2001)
7. Ruiz, A., Price, Y.W.: Test-driven GUI development with testng and abbot. IEEE Software 24(3), 51–57 (2007)
8. Google Inc. Android activity, (2011), <http://developer.android.com/reference/android/app/activity.html> (cited 2011-03-09)
9. Google Inc. Testing fundamentals, (2011), http://developer.android.com/guide/topics/testing/testing_android.html (cited 2011-03-09)
10. Reenskaug, T.M.H.: Models - views - controllers (1979), <http://heim.ifi.uio.no/~trygver/1979/mvc-2/1979-12-MVC.pdf> (cited 2011-03-09)
11. Feldman, A., Daymon, M.: WPF in Action with Visual Studio 2008. Manning Publications Co., Greenwich (2008)
12. Feathers, M.: Working Effectively with Legacy Code. Prentice Hall PTR, Upper Saddle River (2004)
13. Google Inc. Android developing introduction (2011), <http://developer.android.com/guide/developing/index.html> (cited 2011-03-09)
14. Pivotal Labs. Robolectric (2011), <http://pivotal.github.com/robolectric/> (cited 2011-03-09)

15. IEEE 1008 - IEEE standard for software unit testing (1987)
16. Freedman, R.S.: Testability of software components. *IEEE Transactions on Software Engineering* 17, 553–564 (1991)
17. Fowler, M.: *Refactoring: Improving the Design of Existing Code*. Addison-Wesley, Boston (1999)
18. Linnenkugel, U., Müllerburg, M.: Test data selection criteria for (software) integration testing. In: *Proceedings of the first international conference on systems integration on Systems integration 1990*, pp. 709–717 (1990), <http://portal.acm.org/citation.cfm?id=93024.93262>

Security Modeling of SOA System Using Security Intent DSL

Muhammad Qaiser Saleem, Jafreezal Jaafar, and Mohd Fadzil Hassan

Department of Computer and Information Sciences, Universiti Teknologi PETRONAS,
31750 Tronoh, Perak Darul Ridzuan, Malaysia
qaiser_saleem73@hotmail.com, jafreez@petronas.com.my,
mfadzil_hassan@petronas.com.my

Abstract. Currently most of the enterprises are using SOA and Web Services technologies to build their web information system. MDA principles are used to develop web service and they used UML as a modelling language for business process modelling. Along with the increased connectivity in SOA environment, security risks rise exponentially. Security is not defined during the early phases of development and left onto developer. Properly configuring security requirements in SOA applications is quite difficult for developers because they are not security experts. Furthermore SOA security is cross-domain and all required information are not available at downstream phases. General purpose modelling language like UML lacks the model elements to define the security requirements of the business processes. As a result, business process expert either ignore the security intents in their model or indicate them in textual way. A security intents DSL is presented as a UML profile where security intents can be modelled as stereotypes on UML modelling elements during the business process modelling. Aim is to facilitate the business process expert in modelling the security requirements along the business process modelling. This security annotated business process model will facilitate the security expert in specifying the concrete security implementation. As a proof of work we apply our approach to a typical on-line flight booking system business process.

Keywords: Service Oriented Architecture, Model Driven Architecture, Business Process Modeling, Security Intents.

1 Introduction

IT-infrastructure have been evolved into an enterprise landscape which is basically a distributed and loosely coupled, Service Oriented Architecture (SOA) environment [1]. In the new business scene, where companies are using intensive use of Information and Communications Technologies (ICT), they are also increasing their vulnerabilities. With the increase in number of attacks on the system, it is probable that an intrusion can be successful [2]. The security violation defiantly cause losses, therefore it is necessary to secure the whole system. If we talk about SOA security then it is not sufficient to just protect a single point, a comprehensive security policy is required [1].

Security must be unified with the software engineering process but in practice it is considered afterthought and implemented in ad-hoc manner [2]. Furthermore it is left to the developer and added when the functional requirements are met or at the time of integration of distributed applications which is not a realistic approach [3]. SOA applications are cross-domain and coupled over various network technologies and protocols; just adding security code to software applications is not a realistic approach because all required security information are not available at the downstream phases[3, 4]. This approach degrade implementing and maintaining security of the system [5].

During the past few years, several SOA security protocols, access control models and security implementations have emerged to enforce the security goals [3, 6]; however focus of the SOA security standards and protocols are towards technological level; which do not provide high level of abstraction and mastering them is also a daunting task [1, 7]. Dealing security only at implementation stage will leads to security vulnerabilities, which justify increasing effort in defining security in pre-development phases, where finding and removing a bug is cheaper [8].

Business process modeling is the most appropriate layer to describe security requirements and to evaluate risks [1]. Business process modeling is normally performed in a modeling language such as Unified Modeling Language (UML) or Business Process Modeling Notation (BPMN). These modeling languages do not support specification of security requirements [9]. Some security extensions are proposed to annotate the business process model with security goals [10, 11] and the work is in progress. [10]

Model Driven Security (MDS) and automatically developed software having security configuration is a topic of interest among the research community and different research groups across the globe are trying to solve the security problems for SOA based applications by presenting MDS Frameworks [3, 6, 9-13].

Business process modeling can be performed from different perspectives; security expert, business analyst and end user perspectives; and at different levels of abstraction [2]. Both experts; business domain expert as well as security expert; work side-by-side while designing a business process model and defining security requirements [6]. Empirical studies shows that those, who model the business process i.e. business domain expert are able to specify security requirements at high level of abstraction [2]. It is evident that business domain expert must define the security requirements at business process model [14]. However in practice, business domain expert mainly focus on the functionality of the system and often neglect the security goals. It may be happened due to many reasons e.g. business domain expert is not a security expert [2] and no currently available process modeling notation have ability to capture security goals[14]. Furthermore system model and security models are disjoint and expressed in different ways i.e. system model is represented in a graphical way in a modeling language like Unified Modeling Language (UML) while security model is represented as a structured text [2]. Incorporating security goals into a business process model is a challenging task due to many reasons [15]:

- There is not a clear identification of security requirements to be modeled.
- Absence of notations to express the security requirements.
- Difficulty in integrating security requirements into business processes modeling.

Our aim is to facilitate business process expert to add security goals while modeling business process for SOA based systems. Security annotative business process model will facilitate the security expert while defining concrete security implementation. In our work:

- We have provided detail analysis of basic security intents for modeling security objectives in a business process model i.e. confidentiality, integrity, availability auditing.
- We have presented a Domain Specific Language (DSL) to express these security requirements. We have used UML-profiling mechanism to extend the UML and proposed security stereotypes.
- As a proof of concept; we have projected our work to a real world business process model.

Being able to express security requirements in a widely used design notation like UML; helps to save time and effort during the implementation and verification of security in system [17].

2 Related Work

We need a language for modeling security during designing the system which provides syntax and semantic as provided by the UML and BPMN. To fulfill the security requirements in modeling languages, different extensions to the modeling languages are proposed. To model the security objectives related to different system's aspects different security extensions are proposed by different authors. Mostly authors represent the abstract syntax of their DSL by a meta-model using MOF framework and concrete syntax by UML profile [2, 8, 12, 18]. Related work exists almost along all type of software development models, following is its descriptions:

System Models: Static structure of the system is represented by UML class diagram and UML state diagram [19]. Basin David et al. in [5] presented SecureUML to model the security requirements for modeling static structure of the system. Basically it is a separate language based on protocol of Role Based Access Control (RBAC). Afterwards SecureUML can be integrated with any system modeling language like UML or BPMN to model the security in the system design. They have presented a meta-model for abstract syntax and used UML profile for concrete syntax and security constraints are added through OCL.

Interaction Diagram: UML sequence diagram is used to represent the flow of control between the object of the system [19]. Jürjens, J. in [20] defined UMLSec by extending the UML and developed a UML profile to incorporate security to represent the secure interaction.

Deployment Diagram: UML component diagram is used for the representation of deployment of a system [19]. UMLSec presented by Jürjens, J. in [20] also support the secure modeling of UML component diagram.

Work Flow Model: UML activity diagram and BPMN are used to represent the business process work flow. This is the most important aspect of a system and most of the security extensions are proposed related to this aspect.

Rodriguez A. et al. created a meta-model for their security extensions and defined security stereotypes and developed a DSL. They also assign different symbols to these security stereotypes. They used the same DSL for extending BPMN in [2] as well as UML in [8]. Christian Wolter et al. in [14], incorporate security stereotypes in BPMN. Ruth Brue et al. in [12] also presents security stereotypes in UML activity diagram.

3 Literature Study

3.1 Service Oriented Architecture (SOA)

SOA paradigm makes the software application development easy by coupling services over intranet and via the Internet [3]. SOA paradigm has changed the Internet from being repository of data to repository of service [10]. SOA is an architectural style in which software applications are comprised of loosely coupled and reusable services by integrating these services through their standard interface. Services are independent of language, platform and location and may be locally developed or requested from the provider. A business process can be realized as a runtime orchestration of set of services. Software applications are often comprised of numerous distributed components such as databases, web servers, computing nodes, storage nodes etc. and these components are distributed across different independent administrative domains. Services are used but not owned by the user and they reside on provider side. The reusability, agility, cost effectiveness and many other attributes of SOA paradigm has attracted the organizations to adopt it for software development [21-23].

The basic building block of a SOA paradigm is a service. “A *service is an implementation of a well-defined piece of business functionality, with a published interface that is discoverable and can be used by service consumers when building different applications and business processes*” [24]. SOA paradigm can be implemented with different technologies like CORBA, Web Services, JINI etc.; however Web services technology is a widespread accepted instantiation of SOA [23, 25].

3.2 Web Services (WS)

Web Services are defined as “*self-contained, modular units of application logic which provide business functionality to other applications via an Internet connection*” [25]. Software applications are developed by integrating different web services either newly built or legacy applications by avoiding difficulties due to heterogeneous platforms and programming languages by exploiting the XML (Extensible Markup Language) and the Internet technologies [25, 26]. Web service enable the dynamic connections and automation of business processes within and across enterprises for EAI (Enterprise Application Integration) and B2B (Business-to-Business) integration using the web infrastructure with relative standards like HTTP (Hyper Text Transfer

Protocol), XML SOAP (Simple Object Access Protocol) WSDL (Web Services Description Language) and UDDI (Universal Description Discovery and Integration).

3.3 Business Process Modeling

Business Process Modeling is gaining more and more attention in an organization because it is the foundation to describe the organizational workflow [1]. An effective business process model will facilitate the stakeholders of the business to understand the different aspects of the business system and provide a platform to discuss and agree on key fundamentals for achieving the business goals [2]. A business process is defined as “*a set of procedures or activities which collectively pursue a business objective or policy or goal*” [2]. It can also be defined as “*a set of activities and execution constraints between these activities*”[1]. Different techniques are used for business process representation; Damij, N. in [27], group them in two categories; diagrammatic and tabular. Christian Wolter et al. in [14] described different popular diagrammatic business process modeling notations like BPMN, UML, XPDL, JpdI; among these UML and BPMN are considered as industry standards [2].

3.4 Model Drive Architecture (MDA) and Model Driven Security (MDS)

Currently software engineering is greatly influenced by a new MDA paradigm which work at model and meta-model level [28]. In MDA approach software systems are specified and developed through models; transformation functions are automatically performed between models at different levels of abstractions as well as between models to code [5]. Model based design methodology is being widely accepted in the development of electronics systems due to their flexibility and tool support. To organize landscape of model, meta-modeling techniques are emerged; theories and methods are provided for the development of coordinated representation suitable for heterogeneous environment such as SOA [29].

MDS specializes MDSD towards information security [30]. MDS is a technology where security requirement are defined as a model during designing phase and concrete security configuration files can be generated by model transformation [4].

4 Organizational Security Goals

Security is an abstract concept which can be defined by specifying a set of security goals. These security goals can be further subdivided, specialized or combined [14]. During our work we mainly focus security measures to encounter the threats related to: use of identity information and associated rights (authentication, authorization), information in different forms i.e. stored, transferred or processed (confidentiality and integrity of data) and service function (availability and integrity of a system) [1].

4.1 Security Objectives in Related Work

Different research groups are focusing on different security goals for their DSLs [2, 3, 8, 14, 31]. In [30] Michal Hafner et al. defined the three security goals naming

confidentiality, integrity and availability. They defined access control as confidentiality and availability is used in the meaning of no-repudiation. In [2, 8] Alfonso Rodríguez et al. extended the UML and BPMN by defining DSLs and focusing on five security goals: access control, integrity, privacy, attack-harm detection and non-repudiation. In [14]. Christian Wolter et al. presented a security policy model by focusing six security goals: authentication, authorization, confidentiality, integrity, availability, auditing. Michal Menzel et al. also used security policy model in their work [1] and defined security extensions to the BPMN. In [4] Yuichi Nakamura et al. defined three security intents for their work: authentication, integrity and confidentiality and defined a UML profile. In [3] Yuichi Nakamura et al. addressed four business level security intents as they are easy to be understood by business user and presentation of them is discussed in UML: Authentication, Integrity, Non-repudiation and confidentiality. Basically they picked some of the security intents defined in [16] and their names are changed according to WS-Security's terminology.

Among the security objectives mentioned above, we believe following are the essential security objectives which should be modeled in a business process model of SOA applications; which are focused by different authors either as it is or with some different name or by merging them.

1. Confidentiality:

It specifies the system's state where only authorized entities can access the information. Access control is maintained by authentication and authorization. Authentication is a mechanism to verify the identity of an entity. Authorization is based on some specific security model, how to grant various privileges to various entities on different resources [30]. Many authors treat confidentiality, authentication and authorization as a separate security goals [1, 2, 8, 14]. However; Ruth Brue and Michal Hafner in their work [30] keep authentication and authorization under the umbrella of confidentiality and we agree with their work because by enforcement of these access control mechanism one can achieve confidentiality.

2. Integrity:

It identifies an authorized subject to alter information in authorized ways. It ensures the integrity of data (properness of information) as well as integrity of origin [30]. Transferred, processed or stored data can only be modified with proper rights [14]. Basically it ensures that the transferred data between parties must be guaranteed to reach the recipient in the same form and with the same content [3].

3. Availability:

It is an important aspect of reliability and in SOA environment, it is interpreted as non-repudiation. A user may use a resource or call a service and this usage or service call must not deniable. Basically it is a system state where provision of a specific resource is guaranteed [30, 31]. It ensures that the information must include the digital signatures of the parties related to the document [3].

4. *Traceability and Auditing:*

It is a process of verification of all actions performed in an information processing system [14]. It underlies each security requirement and will automatically be understood when a security requirement is specified in a model [2], as there is no need to model it separately in a business process model.

5 **Extending a Modeling Language According to a Particular Domain: Domain Specific Language (DSL)**

General purpose modeling languages like UML are very successful and they also provide the tool support ranging from requirement engineering to code generation. However they does not render the superfluous of DSLs; furthermore it is very clumsy for tasks that can benefits from the integration of the domain-specific restrictions [18]. DSLs are small and provide basis for domain-specific formal analysis; furthermore DSLs use those notions which are familiar to domain experts [18]. DSL is used to formalize a modeling language capable of formalizing different business domains (like e-government, e-education), system aspects (like security, real-time) or concrete technologies(such as EJB or .NET [5]). Extending a modeling language according to a particular domain and defining DSL is a common practice e.g. UML extensions according to specific domains like data warehousing[31], Business intelligence[32] and real-time systems [29]. Following are the three alternatives for defining a DSL [5, 33].

1. The easiest way of defining a DSL is the usage of the extensions points provided by the language itself [33]. DSL can be defined directly in UML in a lightweight way by using stereotypes and tagged values known as “labels” resulting *UML profile*. To introduce new language primitives (elements), *stereotypes* are used by extending the semantics of existing types in the UML meta-model. Stereotypes are represented by double angle brackets e.g. <<*stereotype*>>. To formalize the properties of these new language primitive, *tagged values* are used which are written within curly brackets e.g. {Tag, Value} [34], which associate data with model elements. Model elements are assigned to these new language primitives and labeled them with corresponding stereotype. If some additional restrictions are required on the syntax of these new language primitives; Object Constraints Language (OCL) constraints is used. OCL is a specification language provided by UML, based on first order logic. Normally OCL expressions are used for various purposes such as invariant for classes, pre and post conditions for methods and guards for state diagram. Set of such definitions i.e. stereotype, tagged values and OCL constitutes the UML profile [5].

Most of the currently available UML modeling tools can readily be used because they support the definition of custom stereotypes and tagged value. Because of having tool support this approach is widely used [5, 18, 20]. Normally DSLs are defined by UML-Profiles when the “domain” may be combined with other domains, in an unpredictable way and the model defined under the domain may be interchanged with other domains [18].

It is very clumsy to add domain-specific restrictions in large languages like UML; furthermore for formal analysis, large languages usually lack detailed formal semantics [18]. Visualization of the complicated security intents might be confusing; furthermore, many modeling languages do not provide extension points [33].

Remaining two extension techniques are meta-model based techniques and known as heavy weight extension mechanism. Meta-model based technique of defining DSL is mostly used when the “domain” is well defined and has accepted set of concepts; there is no need to combine the domain with other domains and the model defined under the domain is not transferred into other domains [18].

2. DSL can be defined by using MOF by extending the meta-model of existing modeling languages like UML. Concept of stereotype is used to formally extend the meta-model of an existing modeling language. At modeling level, stereotypes are manipulated as annotation on model elements. In this way of DSL definition, an existing meta-model is reused and specialized.

Limitation is that the extended and customized meta-model is based on the entire meta-model of existing modeling languages and may be complex. Furthermore to support the DSL; CASE (Computer Aided Software Engineering) tool may also require extension to accommodate these new language primitives in particular storage component (repository) and visualization component [5, 18, 29]. Furthermore; extensions are defined and integrated according to a particular domain into a specific modeling language based on its meta-Model [33].

3. A new DSL for modeling the domain of interest or particular problem is created by a fully dedicated meta-model using MOF having no dependency on existing modeling languages. The resulting DSL have much more concise vocabulary than the vocabulary of existing modeling languages e.g. UML. For querying and manipulating meta-data of these DSL, interface would be more simple than the UML Interfaces. Abstract syntax is represented by the meta-model and notions (concrete syntax) of the DSL are specified with the UML profile [5]. This way of extension is optimally suited for the problem at hand [29].

Limitation is, sometime it does not provide the well-defined mapping between the UML model with which developer work, to the instances of meta-model of DSL that define the meaning of this model [18].

To gain the benefits of DSL and general purpose modeling language, DSLs are defined in terms of general purpose modeling language like UML or BPMN [18]. Current practice of defining a DSL by different researchers [2, 5, 8, 14, 20] is; abstract syntax is represented by a meta-model and concrete syntax (notion) is represented by a UML profile. We are also working along this approach.

6 Proposed Domain Specific Language

To gain the benefits of DSL and general purpose modeling language, DSLs are defined in terms of general purpose modeling language like UML or BPMN [23]. In

our research work our domain is “*modeling the security in SOA system*”. General purpose modeling language like UML can easily be customized by the extension mechanism provided by the language itself and DSL can be defined according to the domain of interest by extending the general purpose modeling language. In case of UML the extension mechanism is known as *UML Profile*. Tools are available for the general purpose modeling languages which support the definition and usage of DSL. In our case we have focus the domain of “*SOA Security*” and we have extended the general purpose modeling language UML by providing a DSL. We have used MagicDraw tool for UML modeling which support the definition and usage of DSL. The whole phenomenon can be explained by the Figure 1.

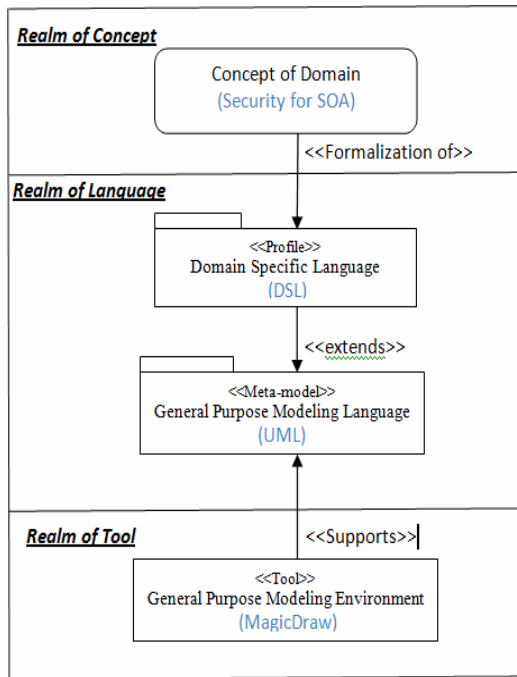


Fig. 1. Definition Process of a Domain Specific Language [29]

Abstract syntax of our DSL is defined by a meta-model and concrete syntax by providing stereotypes. Afterwards UML profiling mechanism is used to apply our DSL into UML.

Each extension of the elements of UML meta-model is formally captured under the concept of stereotypes. Properties and/or modeling constraints of the target domain are associated with the stereotypes which results the UML profile. The most difficult task is the identification of elements of the meta-model of a modeling language which must be extended i.e. in case of UML, identification of UML meta-classes for which

the stereotypes will be defined. In our case we are extending UML meta-class Object-Node. After the definition of domain specific UML-profile, general-purpose modeling tool can easily be specialized and these domain specific stereotypes are made available at the modeling level in the form of annotation [29]. Figure 3 explain the whole concept.

6.1 Abstract Syntax

Abstract syntax of our DSL is presented by a met model. The UML profile that describes our met model is described as UML package with the stereotype <<profile>> as shown in Figure 2. We are using package for the creating of our DSL as discussed in [53]. Our DSL is based on the security intents disused in previous section. The most difficult task is the identification of elements of the meta-model of a modeling language which must be extended for example in case of UML, identification of UML metaclasses for which the stereotypes will be defined [33]. In our case we have extended UML meta-classes *ObjectNode* and *ActivityNode* i.e. these are the metaclasses to which stereotypes will be assigned.

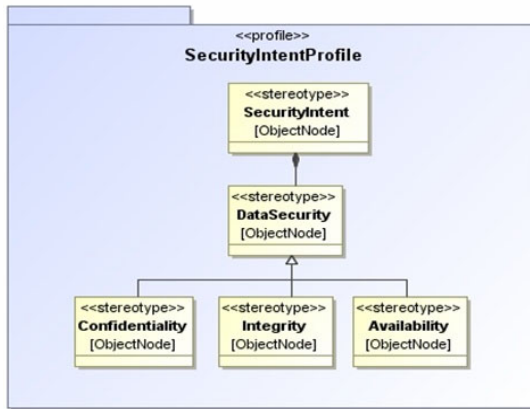

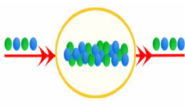



Fig. 2. Abstract Syntax of Proposed DSL

6.2 Concrete Syntax

Each extension of the elements of UML meta-model is formally captured under the concept of stereotypes. Properties and/or modeling constraints of the target domain are associated with the stereotypes which results the UML profile. After the definition of domain specific UML-profile, general-purpose modeling tool can easily be specialized and these domain specific stereotypes are made available at the modeling level in the form of annotation [33]. For concrete syntax we have presented following stereotypes as shown in Table 1.

Table 1. Concrete Syntax (Notions) of Proposed Domain Specific Language

S/No	Security Stereotype	Symbol	Description
1.	<<Confidentiality>>		<p>Idea behind the symbol is that, initially information are inaccessible to user and will only be access able to him/her when he/she provides the desired security credentials. In BPD it can be specified in Pool, Lane, Activity or Group. Idea is to restrict the access to authorized user only.</p>
2.	<<Integrity>>		<p>Idea behind the symbol is that before transformation, information contents are in particular form; during transformation it may change its form however it must be in the same form on its receipt. In BPD it is specified over the Message Flow</p>
3.	<<Availability>>		<p>Basically it is based on the idea of no-repudiation i.e. whenever a user uses some resource or service then his/her signature will be stored with the document along with date and time information. In BPD it can be specified over the message flow, it means it means the interactions cannot be denied.</p>

7 Case Study

To demonstrate our work, a case study of “*Online Flight Booking System*” is presented. It describes the web services based interaction between the participants and enables them to work through the Internet. The whole process has to be realized in a peer-to-peer fashion and would integrate security requirements.

7.1 Business Scenario

In today’s era travel agencies provide online services to travelers for booking the flights. Traveler submits the trip information to the travel agency, containing the personal information of travelers; start date, end date, origin, destination and price rang etc. After having this information travel agency search for the suitable airline and routes accordingly and prepare itinerary and send it to traveler. If traveler accepts

the itinerary then he/she will make payment into the bank specified by the travel agency. The bank; upon receiving payment send receipt of payment to both i.e. traveler as well as travel agency. After receiving conformation of payment, travel agency will order ticket from airline, which will send the ticket to the traveler.

7.2 Stakeholders

In the case-study; services from the four stakeholders are involved i.e. traveler, travel agency, airline and bank.

7.3 Security Requirements of the System

In online flight booking system a traveler needs to perform different tasks i.e. fill in the trip information form, viewing the itinerary, make payment into the bank, view the ticket etc. Necessary permissions are assigned to him/her on different objects to perform these tasks i.e. travels require update information on trip information payment form, read permission on itinerary information and ticket. To perform these operations traveler's personal information are involved at different places e.g. passport number while filing the trip information, credit card information while making payment to bank etc. Therefore *confidentiality* is required i.e. proper *access control* mechanism with *authentication* and *authorization* is required to access this information. Furthermore, traveler has to submit the trip order to the travel agency, traveler must sign it with his/her signature so he/she may not be able to deny that he/she has not submitted the trip order. *Availability (Non-repudiation)* is required in this use-case between the traveler and travel agency. Travel order form is submitted online, therefore secure information flow i.e. *Integrity* is required to successfully perform this use-case. These three security requirements i.e. *Confidentiality*, *Availability*, and *Integrity* are identified and modeled for other stakeholders of the case-study like travel agency, airline and bank. Figure 6 shows the security enhanced business process model of the flight booking system use case.

Meaning of a particular security symbol at a specific place is discussed below.

Confidentiality: Whenever some information are sent or received they are consider as confidential i.e. we show confidentiality requirement on data objects.

Availability (Non Repudiation): Whenever some information would be sent or received between the stakeholders; then availability security requirements would be modeled to ensure the non-repudiation. It represents that sending person would include additional information like digital signature, time and date along with the message, so the interactions cannot be denied.

Integrity: This security requirement is modeled whenever some transmission of information is takes place. It represents integrity of the information transmitted over the Internet. In the case study whenever stakeholders interact with each other through sending messages; integrity symbols would be modeled over the message flow to ensure the integrity of information flow.

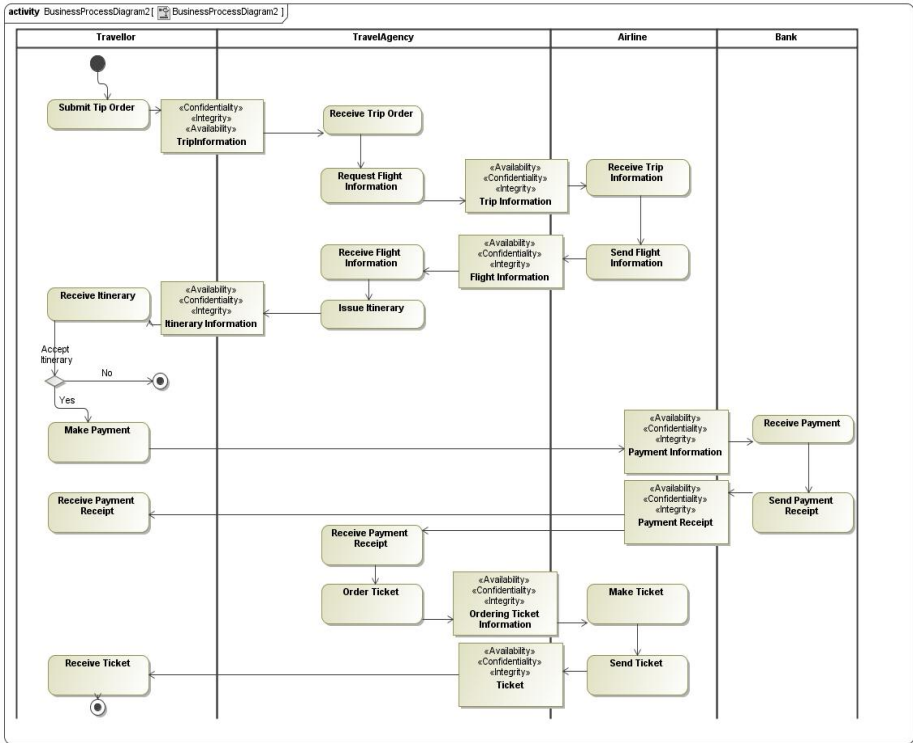


Fig. 6. Security Annotated UML Activity Diagram (Business Process Model) of the case study

8 Conclusion and Future Work

Incorporating security requirements during early stages of software development will improve the important aspect “Security” of SOA based Information Systems. A security DSL is presented to model the security along with the business process model. We have facilitated the business process expert in modeling the security requirements along with the business process model. This security annotated business process model will facilitate the security expert in specifying concrete security implementation. We believe our effort is a contribution towards stressing to incorporate security requirements during business process modeling for SOA applications.

We are in the process of enhancing our DSL to incorporate more security intents which are essential to be modeled during business process modeling for SOA applications.

References

1. Menzel, M.T., Meinel, I.C.: Security Requirements Specification in Service-Oriented Business Process Management. In: International Conference on Availability, Reliability and Security, 2009. ARES (2009)
2. Rodríguez, A., Piattini, E.F.-M.M.: A BPMN Extension for the Modeling of Security Requirements in Business Processes. *IEICE - Trans. Inf. Syst.* E90-D(4), 745–752 (2007)
3. Nakamura, Y.T., Imamura, M., Ono, T. K.: Model-driven security based on a Web services security architecture. In: IEEE International Conference on Services Computing (2005)
4. Satoh, F.N., Mukhi, Y., Tsubori, N.K., Ono, M.K.: Methodology and Tools for End-to-End SOA Security Configurations. In: IEEE Congress on Services - Part I (2008)
5. David Basin, J.D., Lodderstedt, T.: Model driven security: From UML models to access control infrastructures. *ACM Trans. Softw. Eng. Methodol.* 15(1), 39–91 (2006)
6. Christian Wolter, M.M., Meinel, C., Schaad, A., Miseldine, P.: Model-driven business process security requirement specification. *J. Syst. Archit.* 55(4), 211–223 (2009)
7. Alam, M.: Model Driven Security Engineering for the Realization of Dynamic Security Requirements in Collaborative Systems. In: Models in Software Engineering, pp. 278–287 (2007)
8. Rodríguez, A., Fernández-Medina, E., Piattini, M.: Towards a UML 2.0 Extension for the Modeling of Security Requirements in Business Processes, in Trust and Privacy in Digital Business, p. 51-61 (2006)
9. Menzel, M.M.: A Security Meta-model for Service-Oriented Architectures. In: IEEE International Conference on Services Computing, SCC 2009 (2009)
10. Jurjens, J.: UMLsec: Extending UML for Secure Systems Development- Tutorial. In: Proceedings of the 5th International Conference on The Unified Modeling Language. Springer, Heidelberg (2002)
11. Torsten Lodderstedt, D.A.B., Doser, J.: SecureUML: A UML-Based Modeling Language for Model-Driven Security. In: Proceedings of the 5th International Conference on The Unified Modeling Language. Springer, Heidelberg (2002)
12. Michal Hafner, R.B., Agreiter, B.: SECTET: an extensible framework for the realization of secure inter-organizational workflows. *Emerald Internet Research* 16(5), 491–506 (2006)
13. Mukhtiar Memom, M.H., Breu, R.: SECTISSIMO: A Platform-independent Framework for Security Services. In: MODSEC 2008 Modeling Security Workshop (2008)
14. Wolter, C., Menzel, M., Meinel, C.: Modelling Security Goals in Business Processes. In: Proc. GI Modellierung 2008, March 2008, GI LNI 127, pp. 197–212. Berlin, Germany (2008)
15. Baresi, L., et al.: Incorporating Security Requirements into Service Composition: From Modelling to Execution. In: Service-Oriented Computing, pp. 373–388. Springer, Heidelberg (2009)
16. Johnston, S.: Modeling security concerns in service-oriented architectures. IBM developerWorks (2004)
17. Jurjens, J.: Developing Secure System with UMLsec From business process to implementation. Computing Laboratory University of Oxford GB (2001)
18. Achim, D., Brucker, J.u.D.: Metamodel-based UML Notations for Domain-specific Languages. In: 4th International Workshop on Language Engineering (atem 2007), p. 1 (2007)
19. Mikael Åkerholm, I.C.: Goran Mustapić Introduction for using UML (2004)

20. Jürjens, J.: UMLsec: Extending UML for Secure Systems Development. In: UML — The Unified Modeling Language, pp. 1–9 (2002)
21. Lewis, G., Morris, A., Simanta, E., Wrage, S.: Common Misconceptions about Service-Oriented Architecture. In: Sixth International IEEE Conference on Commercial-off-the-Shelf (COTS)-Based Software Systems, ICCBSS 2007 (2007)
22. Asit Dan, P.N.: Dependable Service-Oriented Computing. IEEE Internet Computing 2009, 11–15 (March/April 2009)
23. Philip Bianco, R.K., Merson, P.: Evaluation of Service-Oriented Architecture. Software Engineering Institute/ Carnegie Mellon, 2007. Technical Report, CMU/SEI-2007-TR-015 (September 2007)
24. O'Brien, L., Bass, L., Merson, P.: Quality Attributes and Service-Oriented Architectures Software Engineering Institute/ Carnegie Mellon, Technical Note: CMU/SEI-2005-TN-014 (September 2005)
25. Bucchiarone, A., Gnesi, S.: A Survey on Services Composition Languages and Models. In: International Workshop on Web Services Modeling and Testing, WS-MaTe 2006 (2006)
26. van der Aalst, W.M.P., Dumas, M., ter Hofstede, A.H.M.: Web service composition languages: old wine in New bottles? In: Proceedings of The Euromicro Conference (2003)
27. Damij, N.: Business process modelling using diagrammatic and tabular Techniques. Business Process Management Journal 13(1), 70–90 (2007)
28. Rodríguez, A., Fernández-Medina, E., Piattini, M.: Towards CIM to PIM Transformation: From Secure Business Processes Defined in BPMN to Use-Cases. Business Process Management, 408–415 (2007)
29. Passerone, R.D., Ben Hafaiedh, W., Graf, I., Ferrari, S., Mangeruca, A., Benveniste, L., Josko, A., Peikenkamp, B., Cancila, T., Cuccuru, D., Gerard, A., Terrier, S., Sangiovanni-Vincentelli, F.: Metamodels in Europe: Languages, Tools, and Applications, vol. 26(3), pp. 38–53. Copublished by the IEEE CS and the IEEE CASS (2009)
30. Michal Hafner, R.B.: Security Engineering for Service-Oriented Architectures. Springer, Heidelberg (2009)
31. Luján-Mora, S., Trujillo, J., Song, I.-Y.: Extending the UML for Multidimensional Modeling. In: Jézéquel, J.-M., Hussmann, H., Cook, S. (eds.) UML 2002. LNCS, vol. 2460, pp. 265–276. Springer, Heidelberg (2002)
32. Stefanov, V., List, B., Korherr, B.: Extending UML 2 Activity Diagrams with Business Intelligence Objects, In: Data Warehousing and Knowledge Discovery, p. 53-63 (2005)
33. Menzel, M., Meinel, C.: SecureSOA Modelling Security Requirements for Service-Oriented Architectures. In: IEEE International Conference on Services Computing (SCC) (2010)
34. Saleem, M.Q., Jaafar, J., Hassan, M.F.: Model Driven Security Frameworks for Addressing Security Problems of Service Oriented Architecture. In: International Symposium in Information Technology, ITSIm (2010)

An Input-Driven Approach to Generate Class Diagram and Its Empirical Evaluation

Faridah Hani Mohamed Salleh

Department of Software Engineering,
College of IT, Universiti Tenaga Nasional,
43009 Kajang, Selangor, Malaysia
faridahh@uniten.edu.my

Abstract. This paper presents an approach for generating Unified Modeling Language (UML) class diagram. The objective of the project is to assist users in generating class diagram in a systematic way, with less dependent to the developers' skill. The approach consists of a number of steps or instructions to be executed by users. Executing the steps requires the user to enter inputs that will then generate half-completed class diagram. Step-by-step approaches are presented to show how the candidate of classes, attributes and relationships between classes are captured. The proposed approach is presented using a case study of ATM system. Testing was conducted to assess the effectiveness of the proposed approach. Analysis and discussion of the testing results are discussed in a few last sections of this paper. The development of the system is underway.

Keywords: Object-oriented analysis and design, class diagram.

1 Introduction

Class diagram is categorized as an important diagram in the development of object-oriented application. Class diagram is often starting to be discovered during analysis phase. A process of generating a complete and correct class diagram is something that needs to be put into consideration. For the large and complex system, the process of generating the class diagram can be very time consuming and in many cases, it would result in many problems if it is not being done carefully. Should a class be badly identified by developers, it can complicate the application's logical structure, reduce reusability, and deter the application's maintenance [1].

There is no formal or standardized ways of designing OO (object-oriented) application, where developers define and design the application based on their experiences and skills level. Motivated by the above situation, the project attempts to propose an approach that can facilitate class diagram generation, lower the risk of producing an inaccurate class diagram and produce class diagram that useful for development. The real system that simulates the proposed approach is yet to be developed. Thus, the testing was conducted using Excel to mimic the flow of the proposed approach.

At this stage, the approach is designed to be able to find the following elements of class diagram:

- Classes
- Attributes of classes
- Relationships between classes

This proposed approach works based on the following assumptions:

- A complete and correct use case diagram must be produced beforehand.
- Class diagram produced from this approach must go through several refinement processes before it becomes correct and complete.
- The user knows what they want the system to do and manages to list basic flows of system.
- Have knowledge of the application domain.

Method to identify operations and multiplicity are out of this project scope and will be considered for future enhancements of this project.

2 Motivation

The next generation of software engineering will involve designing systems without using paper-based formats, instead using software to develop software [3]. Although most of the techniques were well-explained, up to the author's knowledge, none of them is computerized. There is a shortage of good tools for supporting OO development efforts. The tools include [3]:

- programs to assist in the design of objects,
- manage libraries of reusable objects,
- design and maintain data-input forms and reports and
- coordinate the development efforts of large teams of programmers.

Most of the past and current research projects in OO give attention to the results after execution of OOAD (object-oriented analysis and design) phase. For example, concentration had been given to the resulting class diagram where the diagram was checked for its quality, consistency and many other aspects. Lack of research is conducted to help developers in performing early stage of OOAD where producing useful models for OO development is needed.

The process of generating OO related work products or specifically class diagram cannot be formalized, but relies on experience, creativity and intuition. This situation causes problem to novice developers as they need much time to learn OO concepts compared to expert. Most of the techniques to produce class diagram requires significant training, practice, intuition, and experience, which usually takes at least 6 months of on-the-job training [4]. Since OO technology requires an entirely different approach to software development, there are significant costs associated with making the transition to OO technology. Education and training for developers can be costly and developers can experience a loss of productivity as they learn to work effectively with the new concepts and techniques.

3 Research Methodology

To run project as efficiently as possible, a thorough research or also called literature study was performed. Core issues for investigation include:

- State-of-the-art research in OOAD
- Issues and challenges in adoption of OO in software development industry
- Detail study on concept of UML class diagram and its implementation in development
- Existing approaches in identifying classes and other components of class diagram
- Designing and conducting experiment

Algorithm analysis and design. Knowledge gained from previous study was used to design the approach. Low-fidelity prototype was developed to simulate how the approach works.

Empirical evaluation. Experiment was conducted to assess the efficiency of the proposed approach. Prior to the real testing or experiment, pilot testing was conducted to assess the experiment materials and flows. Then, the pilot test results were analyzed to improve the experiment materials and flows. The experiment was then conducted by comparing class diagram generated from the proposed approach to the class diagram that had been successfully used to develop system. The results were analyzed again but this time, to improve the proposed approach. Fig.1 presents the research methodology of this project.

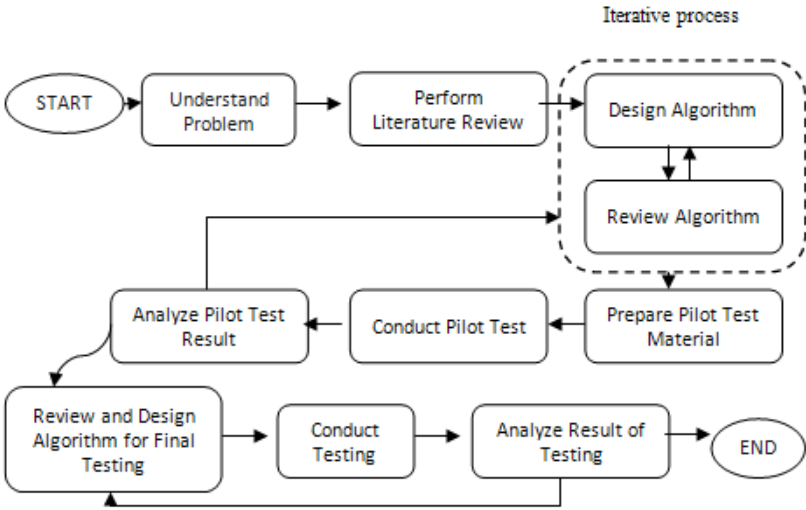


Fig. 1. Research methodology

4 Case Study: ATM

ATM is chosen as a case study to illustrate the implementation of the proposed design. The requirements statement of ATM System is excerpted from [4]:

The software to be designed will control a simulated automated teller machine (ATM) having a magnetic stripe reader for reading an ATM card, a customer console (keyboard and display) for interaction with the customer, a slot for depositing envelopes, a dispenser for cash (in multiples of \$20), a printer for printing customer receipts, and a key-operated switch to allow an operator to start or stop the machine. The ATM will communicate with the bank's computer over an appropriate communication link.

The ATM will service one customer at a time. A customer will be required to insert an ATM card and enter a personal identification number (PIN) - both of which will be sent to the bank for validation as part of each transaction. The customer will then be able to perform one or more transactions. The card will be retained in the machine until the customer indicates that he/she desires no further transactions, at which point it will be returned - except as noted below. The ATM must be able to provide the following services to the customer:

- *A customer must be able to make a cash withdrawal from any suitable account linked to the card, in multiples of \$20.00. Approval must be obtained from the bank before cash is dispensed.*
- *A customer must be able to make a deposit to any account linked to the card, consisting of cash and/or cheques in an envelope. The customer will enter the amount of the deposit into the ATM, respondent to manual verification when the envelope is removed from the machine by an operator. Approval must be obtained from the bank before physically accepting the envelope.*
- *A customer must be able to make a transfer of money between any two accounts linked to the card.*
- *A customer must be able to make a balance inquiry of any account linked to the card.*
- *A customer must be able to abort a transaction in progress by pressing the Cancel key instead of responding to a request from the machine.*

The ATM will communicate each transaction to the bank and obtain verification that it was allowed by the bank. Ordinarily, a transaction will be considered complete by the bank once it has been approved. In the case of a deposit, a second message will be sent to the bank indicating that the customer has deposited the envelope. (If the customer fails to deposit the envelope within the timeout period, or presses cancel instead, no second message will be sent to the bank and the deposit will not be credited to the customer.)

If the bank determines that the customer's PIN is invalid, the customer will be required to re-enter the PIN before a transaction can proceed. If the customer is unable to successfully enter the PIN after three tries, the card will be permanently retained by the machine, and the customer will have to contact the bank to get it back.

If a transaction fails for any reason other than an invalid PIN, the ATM will display an explanation of the problem, and will then ask the customer whether he/she wants to do another transaction. The ATM will provide the customer with a printed receipt for each successful transaction, showing the date, time, machine location, type of transaction, account(s), amount, and ending and available balance(s) of the affected account ("to" account for transfers).

The ATM will have a key-operated switch that will allow an operator to start and stop the servicing of customers. After turning the switch to the "on" position, the operator will be required to verify and enter the total cash on hand. The machine can only be turned off when it is not servicing a customer. When the switch is moved to the "off" position, the machine will shut down, so that the operator may remove deposit envelopes and reload the machine with cash, blank receipts, etc.

The ATM will also maintain an internal log of transactions to facilitate resolving ambiguities arising from a hardware failure in the middle of a transaction. Entries will be made in the log when the ATM is started up and shut down, for each message sent to the Bank (along with the response back, if one is expected), for the dispensing of cash, and for the receiving of an envelope. Log entries may contain card numbers and dollar amounts, but for security will never contain a PIN.

5 The Proposed Approach

The proposed approach uses ‘Input-Driven Approach’, where the diagram is generated based on the answers or responses (input) given by the users through a set of questions and instructions. The answers will then be used to generate the class diagram components. In this design, there are five stages involved, which are presented as follows:

5.1 Identifying Classes and Attributes

- 5.1.1 *Extract nouns from the requirements document.*
- 5.1.2 *Categorize nouns into Things, Event, People and Other category.*
- 5.1.3 *List flow of events for each of the use cases.*
- 5.1.4 *List attributes for each of the listed events.*
- 5.1.5 *Identify the most suitable class for each of the attributes by selecting the class from the categories.*

5.2 Identifying Relationships

1. Use table checking to capture possible associations. Fit the answers obtained from steps described in section 5.1 in the following sentence:
What does a (an) _____ to a (an) _____?
2. Find verb that can describe the connection between the two classes. The verb serves as a role name.
3. Perform refinement to improve relationships.
4. Add new attributes (if any).
5. Regardless the correctness of the attributes, classes that have more than 2 similar attributes have potential to have abstract class that might be used for generalization. Users may think of a class that can be the parent/generalized class.
6. Repeat step 1 but this time use “is_part_of” checking to capture aggregation.

The following section discusses steps listed in section 5.1 and 5.2 in detail.

5.2.1 Extracting Nouns from Requirements Document

Firstly, the nouns are extracted from the requirements statement using NLP (Natural Language Processing).

Requirements Statement for Example ATM System

The software to be designed will control a simulated automated teller machine (ATM) having a magnetic stripe reader for reading an ATM card, a customer console (keyboard and display) for interaction with the customer, a slot for depositing envelopes, a dispenser for cash (in multiples of \$20), a printer for printing customer receipts, and a key-operated switch to allow an operator to start or stop the machine. The ATM will communicate with the bank's computer over an appropriate communication link. (The software on the latter is not part of the requirements for this problem.)

The ATM will service one customer at a time. A customer will be required to insert an ATM card and enter a personal identification number (PIN) - both of which will be sent to the bank for validation as part of each transaction. The customer will then be able to perform one or more transactions. The card will be retained in the machine until the customer indicates that he/she desires no further transactions, at which point it will be returned - except as noted below.

The ATM must be able to provide the following services to the customer:

1. A customer must be able to make a cash withdrawal from any suitable account linked to the card, in multiples of \$20.00. Approval must be obtained from the bank before cash is dispensed.
2. A customer must be able to make a deposit to any account linked to the card, consisting of cash and/or cheques in an envelope. The customer will enter the amount of the deposit into the ATM, subject to manual verification when the envelope is removed from the machine by an operator. Approval must be obtained from the bank before physically accepting the envelope.
3. A customer must be able to make a transfer of money between any two accounts linked to the card.
4. A customer must be able to make a balance inquiry of any account linked to the card.

A customer must be able to abort a transaction in progress by pressing the Cancel key instead of responding to a request from the machine.

The ATM will communicate each transaction to the bank and obtain verification that it was allowed by the bank. Ordinarily, a transaction will be considered complete by the bank once it has been approved. In the case of a deposit, a second message will be sent to the bank indicating that the customer has deposited the envelope. (If the customer fails to deposit the envelope within the limited period, or presses cancel instead, no second message will be sent to the bank and the deposit will not be credited to the customer.)

If the bank determines that the customer's PIN is invalid, the customer will be required to re-enter the PIN before a transaction can proceed. If the customer is unable to successfully enter the PIN after three tries, the card will be permanently retained by the machine, and the customer will have to contact the bank to get it back.

If a transaction fails for any reason other than an invalid PIN, the ATM will display an explanation of the problem, and will then ask the customer whether he/she wants to do another transaction. The ATM will provide the customer with a printed receipt for each successful

Fig. 2. Extracting nouns from the requirements statement

5.2.2 Categorize Nouns into Things, Event, People and Other category

Nouns that are extracted from the requirements statement are listed. Then, based on the users' judgment, categorize the nouns into Things, Event, People or Other category [6]. The descriptions of each of the categories are as follows:

Things: Physical objects

People: Humans who carry out some function

Events: Something that happens at a given place and time

Nouns that cannot be classified under Things, Event or People are categorized under Other category. This step is performed to identify candidate of entity classes. Table 1 shows how each of the nouns is categorized in any of the four defined categories.

Table 1(a). Categorizing nouns into 4 defined categories

Categories			
Things ↓	Event ↓	People ↓	Other ↓
ATM	Approval	customer	account
ATM card	available balance	Operator	amount
bank	balance inquiry		card numbers
bank's computer	communication link		date
cash	deposit		dollar amounts
cheque	display		requirements
customer console	ending balance		security
deposit envelopes	explanation		software
dispenser	hardware failure		time
envelopes	interaction		type of transaction
key operated switch	log entries		
keyboard	request		
machine	timeout period		

Table 1(b). Categorizing nouns into 4 defined categories

Categories			
Things ↓	Event ↓	People ↓	Other ↓
magnetic stripe reader	transaction		
printer	transfer		
receipt	validation		
slot	verification		
internal log	withdraw		
message			

5.2.3 Identifying Events Flow

This step is performed to eliminate irrelevant classes and false attributes. It also serves as a way to trigger the user’s recognition of “potential classes”. Writing list of events is a systematic way to think about how a use case behaves. This can indirectly guide the users to determine what objects or components are needed for the scenario to occur and made sense in the application domain. The users are required to list down all possible events flow or activities for each of the use cases. The possible activities initiated by each of the use cases must be written from actor(s) point of view. Example:

Use case: Withdrawal

Actors: Customer, Bank

Events flow:

1. Customer inserts ATM card into ATM machine
2. Customer enters PIN no
3. Bank verifies PIN no
4. Customer selects withdrawal transaction

From our observation, users tend to overlook the hardware involvement when listing the events flow. Example of activity or event flow that involves hardware:

Customer dispenses money from ATM cash dispenser.

ATM cash dispenser is the hardware that should be included in the event flow. In class diagram, hardware is often to be listed as classes and should be treated as important classes.

5.2.4 Listing the Attributes

The following stage requires the users to identify <<entity>> class and its attributes. There are four types of attributes; descriptive, naming, state information and referential [4]. Identifying state and referential types of attributes is not covered in the proposed approach. Users might not familiar with term ‘attributes’. Thus, the following question is asked to users to guide them in identifying the attributes:

What information should the system keep to perform each of the listed events flow?

Fig. 3 shows the attributes that are identified for use case Withdrawal of ATM system case study. ATM card and card number are some of the attributes identified from the event flow.

Withdrawal		Actors: Customer, Bank					
		1			2		
1	Customer inserts an ATM card with card number into ATM machine.	ATM card	Things		card number	Things	
			Event			Event	
			People	customer		People	customer
			Other			Other	

Fig. 3. Identifying attributes from the listed events flow

5.2.5 Identifying Classes from Identified Attributes

Next, identify the most suitable class for each of the identified attributes. The class can be chosen from any of the four categories. The content of the categories is identified in step 5.1.2. Fig. 4 shows how the attributes and classes are captured.

Withdrawal		Actors: Customer, Bank								
Attributes		1		2		3				
1	Customer inserts an ATM card with card number into ATM machine.	ATM card	Things		card number	Things		ATM machine	Things	Dispenser
			Event			Event			Event	
			People	Customer		People	Customer		People	
			Other			Other			Other	
2	Customer enters PIN number.	PIN number	Things			Things			Things	
			Event			Event			Event	
			People	Customer		People			People	
			Other			Other			Other	
3	Bank verifies the PIN number from customers.	PIN number	Things	Bank	Authentication	Things			Things	
			Event			Event	Validation		Event	
			People			People			People	
			Other			Other			Other	

Fig. 4. Capturing classes, attributes and inheritance

5.2.6 Capturing Associations

Fit each of the classes identified from section 5.1 in the following sentence:

What does a(an) _____ do to a(an) _____?

This method able to capture the existence of relationships only, but not the direction. Direction will be determined by user manually. Matrix table shown in Table 2 shows the process clearly.

Table 2. Capturing the associations and role names

What does a(an) A do to a(an) B ?

Class		B			
		ATM	Account Info	Balance	Customer Console
A	ATM		x	x	contains
	AccountInfo	x		records	displayed on
	Balance	x	recorded in		displayed on
	Customer Console	is installed in	displays	displays	

In Table 2, classes in A and B will be fit in sentence: *What does a(an) (classes in A) do to a(an) (classes in B)?* Section 5.2.2 explains the above step in detail.

5.2.7 Capturing Role Names

There must be a semantic meaning that describes the relationship. Otherwise, the classes are considered not to have relationship.

For example:

What does a Customer Console do to an Account Info?

Answer: Customer Console displays Account Info.

What does a ATM do to a Customer Console?

Answer: ATM contains an account.

From above answers, Customer Console has link with Account Info and ‘displays’ is the role name.

5.2.8 Refinement to Improve Relationship

Refinement process is performed to improve the direction of relationships.

5.2.9 Adding the New Attributes

Additional attributes for classes can be found after the relationships are identified. Adding these new attributes to the identified classes will improve the completeness of the class diagram.

5.2.10 Capturing Generalization

Classes that have similar attributes have high potential to have abstract class that is used for generalization. In this case, try to think of a class that can be the parent/generalized class. As described in 5.2.4, adding the new parent/generalized class will improve the completeness of the class diagram. For example, attribute

amount, *from* and *to* are found in *Withdraw* class and *Transfer* class. Thus, there should be one class that shall serve as parent class for *Withdraw* and *Transfer* class.

Another way to capture the generalization is by fitting the identified attributes in the following sentence: _____ is a _____.

Example: Restaurant manager is a staff.

If the above statement is correct in the context of the application domain, choose Y (yes). Otherwise, choose N (no).

5.2.11 Capturing Aggregation

Keyword '*is part of*' is used as a clue to find aggregation. Table 3 shows how the attributes are checked for aggregation. The identified attributes are fitted in the following sentence: _____ is part of _____.

Example: Customer Console is part of ATM.

If the above statement is true to be applied in the context of the application domain, 'y' is chosen to indicate 'Yes' and 'n' is chosen to indicate 'No'. Results shown in Table 3 produces class diagram (with aggregation), shown in Fig 5.

Table 3. Capturing aggregation

“is part of”

Class	ATM	Account Info	Card Reader	Customer Console	Cash Dispenser	Log
ATM		n	n	n	n	n
AccountInfo	n		n	n	y	n
Card Reader	y	n		n	n	n
Customer Console	y	n	n		n	n
Cash Dispenser	y	n	n	n		n
Log	n	n	n	n	n	

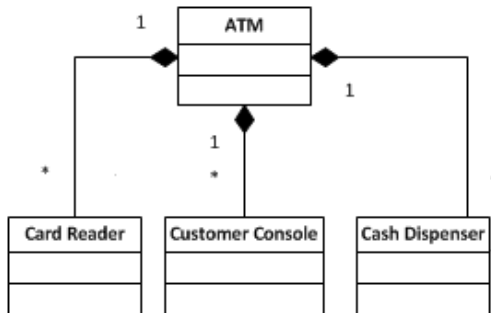


Fig. 5. Result of step to identify aggregation

6 Suggested Refinement Process

The following section explains a few refinement processes that is suggested to be applied when producing the class diagram [2].

Redundant class removal. The users are required to finalize and eliminate any redundant classes derived from the answers of the questions given throughout the process. Redundant class is a class that describes a same meaning to other class. However, to remove the redundant class from the diagram using an automated way is impossible. The tool might face the difficulty to interpret name of the given class name and the actual content of the class. If different words are being used to describe the same idea, selecting the one that is the most meaningful in the context of the system still need to be done manually, on user discretion.

Eliminate irrelevant classes. After executing the entire process, user should also review the identified classes to find for irrelevant classes. We recommend the process of eliminating the redundant classes and refining the remaining classes as proposed by Rebecca Wirfs-Brock, Brian Wilkerson and Lauren Wiener to be used [2].

7 Experimental Design and Setup

The experiment was conducted to three respondents that have background in Computer Science and IT. The objective of the experiment is to assess the efficiency of the proposed approach. None of the respondents has specific background in OOAD. They rely fully on the proposed method to generate class diagram. The respondents took 2.5 to 3 hours to complete the experiment and it was performed individually. Microsoft Excel was used in the experiment to demonstrate the proposed methods. The detail explanation of most of the experiment processes are presented in Section 5 of this paper. The step-by-step experiment processes are as follows:

Step 1: Extracting Nouns. Respondent reads and understands the case study. The case study is presented in Section 4 of this paper. Then, all nouns are extracted from the case study.

Step 2: Categorizing Nouns. The extracted nouns are candidates of entity classes. Respondent categorizes classes into four categories which are *People*, *Things*, *Event* and *Other*.

Step 3: Capturing Classes, Attributes and Inheritance. Respondent lists event flow for each of the main functions of ATM case study. Then, respondent is asked to identify information that needs to be captured by the system to perform each of the listed event flow. Respondent identifies the most suitable noun (class) that holds the information. Fig. 5 shows the screen capture of Excel software that captures the attributes and classes from respondents.

Step 4: Capturing Relationships. Extracted classes from *Step 3* are filled into the matrix table. The processes are described in section 5.2.1, 5.2.2, 5.2.5 and 5.2.6. Fig. 6 presents the screen capture of Excel file used in capturing data for experiment.

Case study: ATM System			
RESPONDENT 001			
Withdrawal	Actors: Customer, Bank		
	1		
1 Customer inserts ATM card into ATM machine	ATM card	Things	<ul style="list-style-type: none"> ATM ATM card bank bank's computer cash cheque customer consol deposit envelope
		Event	
		People	
		Other	

Fig. 6. Capturing classes and attributes

8 Results and Discussion

The classes, attributes and relationships identified from the experiment are compared to the trusted source which is a class diagram of ATM system described in [4]. The results calculation requires distinctive formulas due to elements of class diagram that are connecting to each others. Fig.7 presents the results of experiment. The formula to calculate percentage of result:

$$\text{Result}(\%) = \frac{\text{Total number of identified attributes}}{\text{Total number of attributes}} \times \frac{\text{Total number of identified classes}}{\text{Total number of classes}} \times 100$$

8.1 Analysis of Results and Conclusion

Respondents were able to produce the desired output in two tasks; identifying classes and identifying associations. The most remarkable finding was at identifying association links between classes where two respondents were able to identify 50% of the links between classes. None of the respondents was able to identify classes with inheritance. While result for identifying associations was promising, some modifications need to be done at approach to identify the correct classes, attributes, generalization and aggregation.

Experiment results can be improved by using the real tool that simulates the proposed method to identify class diagram. One of the challenges faced when conducting the experiment was respondents' reluctance in taking part in the experiment. The experiment that requires minimum 2.5 hours to be completed was one of the reasons why most of the potential respondents rejected to be involved in the experiment. The validity of testing results can be improved if more respondents involved in testing the proposed approach. Another findings is, from our observation, most of the respondents executed the steps in testing without reading the guideline. Some mechanism shall be devised to encourage respondents to read guideline before executing the proposed approach to produce class diagram. The respondents shall also be allowed to do the experiment at their own time, pace and place. This will help the respondents to stay focus hence, more accurate inputs can be given. The experiment sessions that were conducted before had only a short break between the sessions.

<i>Identifying Classes & Attributes</i>	RESPONDENT 1	RESPONDENT 2	RESPONDENT 3
Total number of identified classes	12	4	9
Total number of classes	20	20	20
Total number of identified attributes	3	3	6
Total number of attributes	59	59	59
Results (%)	3.05%	1.02%	4.58%

<i>Identifying Relationships</i>	RESPONDENT 1	RESPONDENT 2	RESPONDENT 3
Total number of identified associations/ occurrences	59	8	44
Total number of associations/occurrences	132	12	72
Results (%)	44.70%	66.67%	61.11%
Total number of identified generalization	1	1	1
Total number of generalization	0	0	0
Total classes with aggregation	8	8	8
Total identified classes with aggregations	4	1	3
Total number of identified aggregation	2	0	1
Results (%)	12.50%	0.00%	4.69%

Fig. 7. Results of experiment

In future, extensive evaluation or testing shall be conducted by assessing the proposed approach from various aspects like time taken to generate class diagram, completeness and correctness of class diagram and the most important is the applicability of the resulting diagram to the system development. Each class diagram will be reviewed to ensure that it is well-formed (high cohesion, low coupling). Software quality metrics shall be applied for the review and statistical analysis shall be performed to the experiment results.

8.2 Threats to Validity

The following measures were taken to mitigate the possible threats to data validity as advised by [7]. On construct validity, a pilot test was performed prior to the actual experiment conduction. As a result of the pilot test, a number of steps and instructions were rephrased to ensure that the respondents share common understanding of the instructions given in the experiment. Some other instructions were rearranged to

smoothen the logical flow of the experiment. On external validity, the respondents were chosen carefully to ensure the reliability of the given input. On reliability, the study design and experiment structure were described in sufficient detail.

This experiment requires the respondents' inputs to be compared with one correct class diagram, which was identified by the researcher. Thus, asking the respondents to understand the identified requirements statement is considered as the best method. The validity of the experiment result can be improved by asking the respondents to write down the flow of system that they wish to develop. This method is better than relying on respondents' understanding to the requirements statement completely. Complexity level of requirements statement or case study can be varied, where the respondents may be assessed to produce simple, less simple, moderate, complex and very complex case studies.

9 Conclusion

This project proposes an approach that later produces a design of tool that can assist in generating class diagram, where the dependency to the developers' skill will be reduced and high quality OO application can be produced. The proposed approach also reduces the time taken to generate class diagram and lower the risk of coming out with an inadequate class diagram. The proposed steps and questions are devised to assist the developers or users to produce an accurate class diagram. The proposed design allows users to generate class diagram, where the users will no need to know the technical aspects of the implementation of class diagram. Experiment was conducted to assess the efficiency of the proposed approach. Positive results were shown in two tasks; identifying classes and associations. However, some modifications and enhancements are needed to improve the methods to identify the correct classes, attributes, generalization and aggregation. It is a hope that the output of this project could be a good basis to encourage more usages of OO technology among industry players.

References

1. Mohamed Salleh, F.H., Sulaiman, H., Kasirun, Z.: Comparative Study on Various Approaches of Identifying Classes in Object Oriented Design. In: Proceedings of National Conference on Programming Sciences 2005, Universiti Kebangsaan Malaysia, pp. 230–239 (December 1, 2005)
2. Bahrami, A.: Object Oriented Systems Development: Using the Unified Modeling Language. McGraw-Hill, Singapore (1999)
3. Communications of the ACM- Issue: Object-Oriented Technology: A Manager's Guide 1990, pp. 23. Addison-Wesley Publishing Company David A. Taylor (January 2009)
4. UML and C++: A Practical Guide to Object-Oriented Development, 2nd edn., Richard C.Lee, William M. Tephenthart, Prentice Hall (2001)
5. CPS211.: Object-Oriented Software Development, <http://www.math-cs.gordon.edu/courses/cs211/>
6. Ross, R.G.: Entity Modeling: Techniques and Applications. Database Research Group, Inc. USA, p. 218 (1988)
7. Easterbrook, J., Storey, M.-A., Damian, D.: Selecting empirical methods for software engineering. In: Shull, F., Sjöberg, D.I.K. (eds.) In Guide to Advanced Empirical Software Engineering, pp. 285–311. Springer, Heidelberg (2007)

Understanding Motivators and De-motivators for Software Engineers – A Case of Malaysian Software Engineering Industry

Mobashar Rehman¹, Ahmad Kamil Mahmood¹, Rohani Salleh²,
and Aamir Amin¹

¹Department of Computer Sciences

²Department of Management and Humanities
Universiti Teknologi PETRONAS, Bandar Seri Iskandar,
31750, Tronoh, Perak, Malaysia
mubashir_rehman@yahoo.com

Abstract. One of the key components which has an impact on the performance and productivity of individuals in the organization is motivation. Software engineering lacks the studies on motivation. Even though those studies which have been done so far are mostly from Western countries. This paper tries to fill this gap by understanding motivators and de-motivators in the field of software engineering from Malaysian perspective. Questionnaire method was used to collect the data. Results show that recognition, technically challenging work, job security and feedback are the major motivators in the field of software engineering. As far as de-motivators are concerned, main de-motivators includes lack of promotional opportunities, less competitive pay and unfair reward system. These results confirmed that the importance of motivators and de-motivators vary between Western and Malaysian cultures.

Keywords: Motivators, De-motivators, Software Engineers, Malaysia.

1 Introduction

World economy has moved from industrialization to knowledge economy. Due to this shift, the major portion of Gross Domestic Product (GDP) of many countries in Organisation for Economic Co-operation and Development (OECD) is based on knowledge based activities [1]. Computer software and service industry is one of the examples of knowledge based industries and it is a fast growing industry [2]. Importance of software engineering can be imagined from the findings that this industry is already a leading contributor to world economy and employment [3].

Despite the importance of computer software and service industry, a small number of studies have been done on motivating software engineers. Therefore motivating software engineers continues to be a challenging task [4] because of the less research in this field and incompetence of software engineering managers in social science aspects like motivation. Although, corporate strategic advantage can be achieved through retaining “committed and productive employees” and retention and

motivation are closely related [5]. This employs that motivation plays a vital role in the overall benefit of organization. Question arises that why motivating software engineers is a difficult task? One of the main reasons is the inability of software engineering managers to motivate their sub-ordinates as they do not know about motivational practices [6]. Their background is from technical side and they lack managerial experience. They are promoted to managerial posts based on their skills in technical area and once they become managers or group leaders they have to cope with issues from human behaviour like how to motivate subordinates, how to get maximum output from individuals by keeping their interests in mind. These are not the areas of their (software engineering managers) expertise. Therefore, such kinds of studies are needed in the field of software engineering which should give software engineering managers an insight of motivating software engineers. In other words, studies should be conducted to analyze what motivates and de-motivates software engineers in the field of software engineering. These studies are also needed because motivation increase productivity but it is a difficult concept when it comes to software engineers [6] because software engineering is a distinct group [7], [8].

Not many studies have been conducted on motivating software engineers or analyzing motivators in the field of software engineering in Malaysian environment. Few of the studies which have been done include US and European countries [9], Egypt [10], Finland, Nigeria and Estonia [11], Pakistan [12] and Brazil [13]. As discussed above that software engineering is a distinct profession so there is a need to study how software engineers get motivated or de-motivated as culture of every part of the world is not same. One motivator might be important for software engineers form one part of the world but not for others.

This paper aimed to find out what are the motivators and de-motivators in the field of software engineering for Malaysian software engineers. Do they consider same motivational and de-motivational aspects as other software engineers from other parts of the world? This study will help software engineers and managers in the field of software engineering by provide an insight about motivational aspects in this field.

2 Literature Review

2.1 Motivational Theories and Software Engineering

Software engineering does not have its own motivational theories and till now motivational theories from social science are used in this field. One of the most comprehensive review works on motivation in software engineering was done in [14]. Table 1 show the name of theories and their explanation which are from social science but used in software engineering. These theories can be categorized into process base theories (motivation is described as a process based on more than one activity) and content theories (motivation is measured at a single point of time and is not considered as a process of multiple activities). Among these theories, Job Characteristic Theory and Herzberg Motivational Theory have been used most of the time.

2.2 Motivators for Software Engineers

Demand and turnover both for Management and Information System (MIS) professionals are reported to be high [15]. This higher rate of turnover adversely affects any organization and especially software engineering firms. Reason being that software engineering is a knowledge oriented profession [16] which means that if a software engineer leaves his/her organization, the knowledge possessed by that person will also go with that person. This will leave a gap between required and existing knowledge. Therefore, organization as a whole will suffer in case of higher turnover. It has been reported that de-motivation leads to absenteeism and turnover [9] thus it can be concluded that one of the means to reduce higher turnover is through motivation.

Many motivational aspects have been studied by researchers in the field of software engineering. Like “the need to identify with the task, employee participation/involvement, good management, career path, sense of belonging, rewards and incentive, recognition, technically challenging work, feedback and job security” [9]. Among these motivational aspects, mostly cited is “the need to identify with the task” [9]. This means that if software engineers know well about their job, tasks are clear to them, they know how it benefits customer(s), organization and themselves, they will be motivated to do that kind of job.

Table 1. Various Theories of Motivation

Theory	Explanation	Category
Equity Theory [25]	Balancing individual's input versus output received from organization	Process
Stimulus Response Theory [26]	Individual's behaviour is based on punitive or rewarding stimuli	
Job Characteristic Theory [27]	Work itself is the main motivator which is measured on five dimensions	
Goal Setting Theory [28]	Goals should be set but these goals should be realistic, measurable and feedback should be provided	
Expectancy Theory [29]	Motivation is based on expectation of positive outcomes	
Need Theory – Malsow [30]	During different life stages, motivational needs changes. Hierarchy of needs was presented in this theory	Content
Need Theory – McClelland [31]	Achievement, affiliation and authority are the motivational needs of an individual	Content
Motivation - Hygiene (Herzberg) [32]	Motivational factors were categorized into intrinsic and extrinsic factors	

Source: [14]

Rewards are another important factor for the motivation, a basic factor according to Maslow's Hierarchy of Needs. Besides reward, career path is also important for software engineers. Those organizations, where individuals do not have enough growth opportunities, they will face higher turnover rate as their employees will quit [17].

Problem solving is one more factor discussed by researchers as a motivator in the field of software engineering. As each software is different from other softwares in some aspects based on the customer's requirements, thus software engineers have to deal with different problems while providing solutions to the customers. This leads them to problem solving on regular basis and it is a challenging work. Therefore, technically challenging work is considered as a crucial motivator in the field of software engineering [9].

Job security is also considered as a motivator in the field of software engineering [9]. But the aim is that whether motivational practices in an organization should be individual centric or organization centric? Because, many individuals especially beginners, use organizations as a ladder to move ahead. These people will leave an organization in 3-4 years on average [18]; [19]. Point is, instead of only focusing on job security, organizations should focus on other motivational aspects as well, because those who want to quit the organization in 3-4 years might be retained by giving them other motivational benefits.

2.3 De-motivators for Software Engineers

Just like any other profession in the world, software engineering also has its own de-motivators. Some of them are work and home life imbalance, stress [20], [17], less feedback [21], [10]. Feedback from supervisors and colleagues, especially supervisors is important, otherwise individuals may not know about their performance which can be a cause of de-motivation. Insufficient salary [10], [22] and lacking growth opportunities [22], [23] are also important de-motivating factors.

3 Research Methodology

Personally Administered Questionnaire (PAQ) method was used to collect the data. Simple Random Sampling (SRS) technique was used to select the respondents. SRS was used because results can be generalized after using this method. Data was collected from 80 respondents. Factors mentioned in [9], for motivators and de-motivators in the field of software engineering were used in the questionnaire. These factors were used to compare the results of this study with those done in other parts of the world. Data was analysed using descriptive statistics, frequency method (number of times respondents selected that option).

4 Results and Analysis

Table 2 presents the demographic information. 93.75% respondents were male whereas only 6.25% were female. 50% had 3-4 years of experience while 25% had 1-2 years of experience. Most of the respondents were software developers (43.75%) followed by software testers (37.5%).

Table 2. Demographic Information

Details	Frequency	Percentage
Gender		
Male	75	93.75
Female	5	6.25
Experience		
Less than 1 year	9	11.25
1-2	20	25
3-4	40	50
5-7	11	13.75
More than 7 years	-	-
Job Description		
Software Developer	35	43.75
Software Testing	30	37.5
Software Maintenance	12	15
Software Quality Assurance	2	2.5
Other	1	1.25

Table 3 reports the motivators for software engineers and their frequency, based on the feedback provided by the Malaysian software engineers. As indicated in the research methodology, factors in questionnaire were taken from the study done by [9]. Results show that among the highly reported motivators are recognition (frequency = 32), technically challenging work (frequency = 28), job security (frequency = 26), feedback (frequency = 25), career path (frequency = 24), work/life balance (frequency = 23) and task significance (frequency = 23).

De-motivators for software engineers indicated by Malaysian software engineers are presented in table 4. Among the mostly reported de-motivators are lack of growth opportunities (frequency = 20), insufficient salary (frequency = 19), poor reward system (frequency = 18), no involvement in decision making (frequency = 16), stress (frequency = 14), unrealistic or unachievable goals (frequency = 14) and poor management (frequency = 14).

Table 3. Motivators for Software Engineers

Motivators	Frequency
Recognition	32
Technically challenging work	28
Job security/stable environment	26
Feedback	25
Career Path (opportunity for advancement, promotion prospect, career planning)	24
Work/life balance (flexibility in work times, work location)	23
Making a contribution/task significance (degree to which the job has a substantial impact on the lives or work of other people)	23
Rewards and incentives	20
Autonomy (e.g. freedom to carry out tasks, allowing roles to evolve)	19
Variety of work (e.g. making good use of skills, being stretched)	18
Trust/respect	17
Appropriate working conditions/environment/good equipment/tools/physical space	17
Development needs addressed (e.g. training opportunities to widen skills; opportunity to specialise)	15
Employee participation/involvement/working with others	14
Identify with the task (clear goals, know purpose of task, how it fits in with whole, producing identifiable piece of quality work)	12
Good management (senior management support, good communication)	11
Sufficient resources	11
Empowerment/responsibility (where responsibility is assigned to the person not the task)	10
Sense of belonging/supportive relationships	9
Equity	9
Working in successful company (e.g. financially stable)	6

Table 4. De-Motivators for Software Engineers

De-Motivators	Frequency
Lack of promotion opportunities/stagnation/career plateau/boring work/poor job fit	20
Uncompetitive pay/poor pay/unpaid overtime	19
Unfair reward system	18
Lack of influence/not involved in decision making/no voice	16
Stress	14
Unrealistic goals/ phoney deadlines	14
Poor management (e.g. poorly conducted meetings that are a waste of time)	14
Interesting work going to other parties (e.g. outsourcing)	13
Poor communication (Feedback deficiency/loss of direct contact with all levels of management)	11
Risk	10
Bad relationship with users and colleagues	10
Producing poor quality software (no sense of accomplishment)	10
Poor working environment (e.g., unstable/insecure/lacking in investment and resources; being physically separated from team)	8
Poor cultural fit/stereotyping/role ambiguity	7
Inequity (e.g. recognition based on management intuition or personal preference)	5

5 Discussion

A comprehensive work in studies [9]; [24] concluded that most of the work on motivation in software engineering field is dominated by Western studies. The motivators in the field of software engineering which are important in Western culture are identifying with the task (frequency = 20), employee participation (frequency = 16), good management (frequency = 16), career path (frequency = 15) and rewards (frequency = 14) [9]. Our findings which are based on Malaysian culture indicated that motivators in the field of software engineering from Malaysian software engineer's perspective are different from a study conducted in Western environment. Important motivators for Malaysian software engineers are recognition, technically challenging work, job security, feedback, career path, work/life balance and task significance.

Similarly, there is a difference between importance of de-motivators in the field of software engineering in Western and Malaysian cultures. Studies [9], [24] which summarized results of Western based studies, highlighted the following de-motivators cited most of the times. Poor working environment (frequency = 9), poor management (frequency = 7), uncompetitive pay (frequency = 6), lack of promotion (frequency = 5) and poor communication (frequency = 5). In comparison to this, important de-motivators according to Malaysian culture are lack of growth opportunities, insufficient salary, poor reward system, no involvement in decision making, stress, unrealistic or unachievable goals and poor management.

Thus, a clear deviation can be found between the importance of motivational and de-motivational aspects between Western and Malaysian cultures. This can be explained by the phenomenon that culture has an impact on the individual's characteristics and thus motivators and de-motivators will also vary, based on the characteristics of an individual, a link which was theoretically proved in [9].

6 Conclusion

Software engineering form a distinct group of profession and studies on motivators in software engineering are very few. This study tried to fill this gap by examining the motivators and de-motivators in the field of software engineering from Malaysian software engineer's perspective. Findings suggest that importance of motivators and de-motivators in software engineering field vary from one culture to another. Those motivators and de-motivators which are important for Western software engineers are not necessarily important for Malaysian software engineers.

Although this study gives us some hints about the variations in results from different parts of the world, results of this study cannot be generalized due to low sample size. Thus generalization should be applied after further verification of these results by conducting similar kind of study on a larger scale in Malaysia.

References

1. Organisation for Economic Co-operation and Development (OECD): Scoreboard of Indicators. OECD Paris (1998a)
2. United Nations Conference on Trade and Development: Changing Dynamics of Global Computer Software and Services Industry: Implications for Developing Countries, New York and Geneva (2002),
<http://www.unctad.org/en/docs/psitetebd12.en.pdf>
3. Schwabe, R.: Competing in software. Advanced Technology Assessment System: Information Technology for Development, by UNCTAD, Division on Science and Technology for Development, Geneva (10 Autumn) (1995)
4. Procaccino, J.D., Verner, J.M., Shelfer, K.M., Gefen, D.: What do software practitioners really think about project success: An exploratory study. *J. Syst. Softw.* 78(2), 194–203 (2005)
5. Mak, B.L., Sockel, H.: A confirmatory factor analysis of IS employee motivation and retention. *Information and Management*, 265–276 (2001)
6. Tanner, F.R.: On motivating engineers. In: Engineering Management Conference (IEMC 2003): Managing Technologically Driven Organizations: The Human Side of Innovation and Change, pp. 214–218 (2003)
7. Capretz, L.F.: Personality types in software engineering. *International Journal of Human Computer Studies* 58(2), 207–214 (2003)
8. Ramachandran, S., Rao, S.V.: An effort towards identifying occupational culture among information systems professionals. In: Proceedings of the 2006 ACM SIGMIS CPR conference on computer personnel research: Forty four years of computer personnel research: achievements, challenges & the future, Claremont, California, USA, pp. 198–204. ACM Press, New York (2006)

9. Beecham, S., Baddoo, N., Hall, T., Robinson, H., Sharp, H.: Motivation in software engineering: a systematic literature review. *Information and Software Technology* 50(9-10), 860–878 (2008)
10. Khalil, O.E.M., Zawacki, R.A., Zawacki, P.A., Selim, A.: What motivates Egyptian IS managers and personnel: Some preliminary results. In: *Proceedings of the ACM SIGCPR Conference*, pp. 187–192 (1997)
11. Princely, I.: Motivation and Job Satisfaction among Information Systems Developers – Perspectives from Finland, Nigeria and Estonia: A Preliminary Study. In: Vasilecas, O., Caplinskas, A., Wojtkowski, W.G., Zupancic, J., Wryczw, S. (eds.) *Proceedings of the 13th International Conference on Information Systems: Advances in Theory, Practice Methods, and Education*, Vilnius, Lithuania, Septmebr 9 – 11, pp. 161–172 (2004)
12. Bhatti, M.W., Ahsan, A., Sajid, A.: A Framework to Identify the ‘Motivational Factors’ of Employees; A Case Study of Pakistan IT Industry. *WSEAS transactions on computers* 7(6) (2008)
13. França, A.C.C., da Silva, F.Q.B.: An Empirical Study on Software Engineers ‘Motivational Factors. In: 3rd International Symposium on Empirical Software Engineering and Measurement ESEM, Short Paper Session. Lake Buena Vista, FL USA (2009)
14. Hall, T., Baddoo, N., Beecham, S., Robinson, H., Sharp, H.: A systematic review of theory use in studies investigating the motivations of software engineers. *ACM Transactions on Software Engineering and Methodology* 18(3), 1–29 (2009)
15. Igbaria, M., Siegel, S.R.: The reasons for turnover of information systems personnel. *Information and Management* 23, 321–330 (1992)
16. Bjornson, F.O., Dingsoyr, T.: Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. *Information and Software Technology* 50, 1055–1068 (2008)
17. Carayon, P., Hoonakker, P., Marchand, S., Schwarz, J.: Job characteristics and quality of working life in the IT workforce: the role of gender. In: *Proceedings of the 2003 SIGMIS Conference on Computer Personnel Research: Freedom in Philadelphia—Leveraging Differences and Diversity in the IT Workforce*, Philadelphia, Pennsylvania, April 10 - 12, pp. 58–63 (2003)
18. Agarwal, R., Ferratt, W.T.: Retention and the Career Motives of IT Professional. In: *SIGCPR*, pp. 158–166 (2000)
19. Rousseau, D.M.: New Hire Perceptions of the Their Own and Their Employer’s Obligations: A Study of Psychological Contracts. *Journal of Organizational Behavior* 11, 389–400 (1990)
20. Dittrich, J.E., Couger, J.D., Zawacki, R.A.: Perceptions of equity, job satisfaction, and intention to quit among data processing personnel. *Information & Management* 9(2), 67–75 (1985)
21. Couger, J.D., Adelsberger, H.: Environments: Austria compared to the United States. *SIGCPR Comput. Pers.* 11(4), 13–17 (1988)
22. Santana, M., Robey, D.: Perceptions of control during systems development: effects on job satisfaction of systems professionals. *SIGCPR Comput. Pers.* 16(1), 20–34 (1995)
23. Andersen, E.S.: Never the twain shall meet: exploring the differences between Japanese and Norwegian IS professionals. In: *Proceedings of the 2002 ACM SIGCPR Conference on Computer Personnel Research (SIGCPR 2002)*, Kristiansand, Norway, May 14 - 16, pp. 65–71 (2002)
24. Sharp, H., Baddoo, N., Beecham, S., Hall, T., Robinson, H.: Models of motivation in software engineering. *Information and Software Technology* 51, 219–233 (2009)

25. Adams, J.S.: Toward an understanding of inequity. *Abnormal Social Psych.* 67, 422–436 (1963)
26. Skinner, B.F.: *Walden Two*. Prentice Hall, Upper Saddle River (1976)
27. Hackman, J.R., Andoldman, G.R.: *Motivation Through the Design of Work: Test of a Theory*. Academic Press, New York (1976)
28. Locke, E.A.: Toward a theory of task motivation and incentives. *Organisation Behav. Hum. Perform.* 3, 157–189 (1968)
29. Vroom, V.H.: *Work and Motivation*. Wiley, New York (1964)
30. Maslow, A.: *Motivation and Personality*. Harper & Row, New York (1954)
31. McClelland, D.C.: *The Achieving Society*. Van Nostrand, Princeton (1961)
32. Herzberg, F., Mausner, B., Snyderman, B.B.: *Motivation to Work*, 2nd edn. Wiley, New York (1959)

UML Diagram for Design Patterns

Muhazam Mustapha and Nik Ghazali Nik Daud

National Defense University of Malaysia,
Sungai Besi Camp, 57000 Kuala Lumpur, Malaysia
{muhazam, nikghazali}@upnm.edu.my

Abstract. UML has been used widely as a design and analysis tool for object-oriented software projects. Despite of having a good number of diagram categories, UML is still lacking of a dedicated set of diagrams for representing design patterns. While numerous works have been published on attempts to describe design patterns in terms of UML features, there is hardly any work has been done on attempts to un-clutter class diagrams at the initial development stage so that the design patterns in used can be seen clearly. With the aim to tidy-up huge networks of interconnecting class diagrams, this paper proposes a new higher level set of diagrams and layout schemes to better highlight the design patterns used in a software architecture. This proposal can go further as a proposal for a new set of diagram in UML standard.

Keywords: Design Patterns, Unified Modeling Language (UML), Architecture Description Language (ADL), Object Modeling Language (OML).

1 Introduction

1.1 UML and Design Pattern Background

One can agree that at the highest level of system design, design pattern [12] has been adopted as a way to model and communicate software ideas and architecture [17][29][34] and, more importantly, to start the design idea not-from-scratch [3][21]. Not only that pure software projects can be very conveniently presented in design pattern language like the projects given by Korhonen et al. [20] and Rößling [32], there are also many projects that are not entirely pure software, but have been successfully presented as design patterns. The examples are the parallel computation by Huang et al. [16] and database related projects by Pasala and Ram [28] and Wernhart et al. [37]

One of the most common ways to represent design pattern is through the use of unified modeling language (UML). However, unknown to many, the de facto inventor of design pattern, Gamma et al. did not represent the design patterns in their book [12] in UML format, but in object modeling technique (OMT) instead.

UML has been around since 1997, and as of now it comprises of 14 diagram sets [10][26][27]. Despite the fact that it has been used widely in program development, and despite the fact that it has so many diagrams – that may seem to be redundant – UML is still being regarded as incomplete [10]. To be more specific, in this paper, the

author would like to highlight that, even though design patterns have been represented in UML, there is none of its 14 diagrams is dedicated for design pattern. The author is not the only one making this claim because Riehle, in [31] Section “3 - The Tools and UML Community,” has also made the same statement. Due to this shortcoming, there are numerous works have been done to represent design patterns in form of existing UML diagrams. For example, Dong et al. [4][5][6][7], Le Guennec et al. [14], and many other significant works [33][35][36], have performed extensive studies to achieve such visualization with UML features. The infrastructures that have been used include annotated package diagrams, collaboration diagram, roles, tags, stereotypes, metamodel profiling, as well as Venn-diagram-like dotted or shaded area, object constraint language [25] and Object-Z [19]. Similar works have also been done by France and Kim et al. [11][18] to accommodate the insufficiency of UML class diagrams to represent design pattern through the use of UML sequence and state diagrams.

Not only UML is insufficient for representing design patterns, design patterns themselves are inadequately defined. Many researchers have put a lot of effort to precisely define design pattern like Lauder and Kent [22], Le Guennec et al. [14], Mak et al. [24] and Eden et al. [8][9]. Formal definition of design patterns, not necessarily in form of UML, is vital in many research works. Software reverse engineering projects, like developing tool support for automated recognition and recovery of design patterns [1][23][30][43] and the evaluation of the design pattern itself [15], are all craving for precise definition of design patterns -otherwise the code developed for such projects won't work. The significance of such works become obvious if the code was not well documented or was not developed based on any design pattern.

To complete this background introduction, it is worthwhile to mention one noble example of non-UML scheme to visually and formally specify design pattern, i.e. the one done by Gasparis et al. [13] using LePUS.

1.2 Placing This Paper into the Background

Jakubík [17] proposed almost the same idea of representing design patterns with new kind of diagrams. Besides that, Yacoub and Ammar have used *interface diagrams* in [39][40] to illustrate a way to represent design patterns. These three articles have almost presented the very idea that is to be delivered in this paper. However, their main purpose wasn't to develop a new type of diagram that has potential to be part of UML. This gap that has been left behind by these three closest articles is the one that is to be filled in by this paper.

Besides the three papers, Zdun [41][42] and Buschmann et al. [2] have also presented quite a similar idea but more on expressive languages rather than diagrams.

Other than that, this paper is also focusing on simple, front-end sketch-type diagram that is to be used at inception phase of software development. This, on the other hand, is the gap that the background materials of sophisticated back-end development tools from various literatures in the previous subsection have left behind for this paper.

Dedicated Diagram (Possibly UML) for Design Pattern. As mentioned in previous section, UML has no dedicated diagram for design pattern [31]. As a result, there are many works have been published on attempts to extend or manipulate existing UML features to properly define design pattern. The literatures presented in the previous section show many impressive works to achieve such visual representation, but there is still no work on proposing some proper and dedicated diagrams for design pattern itself.

Filling up this very gap of lacking of dedicated diagram for design pattern is the *first* target of this paper. This diagram can be considered as an architecture description language (ADL) since at the level of design pattern, the scope is architecture. It can also be further proposed to be formally included in UML.

Uncluttered Diagram Layout. The works of Dong et al. in [4][5][7] are very good examples of the attempts to make design patterns stand up in a complicated initial software design. However, the methods used in those literatures have obviously cluttered the entire diagram. The reason is because the very features used to highlight design patterns have added untidiness to the diagram. Contrarily, a good and nicely laid out design diagrams is very vital for the initial stage of software development – inception phase.

Solving this cluttering problem is the *second* target of this paper. This will be achieved by equipping the dedicated design pattern diagram mentioned in the previous section with a set of layout and notation schemes whose main purpose is to reduce the cluttering.

2 Groups of GoF's Design Patterns

To achieve the target in Subsection 1.2 (dedicated design pattern diagram), this paper first has to advocate the design patterns given by GoF¹ in [12] as *elementary design patterns*. It is assumed that most of more recently discovered design patterns would be a composition of these elementary patterns. Should there be a genuinely new elementary design pattern, it should be published and would be added to the collection in Figure 8 - 9.

In order to better understand the way to convert the GoF's elementary patterns to the corresponding dedicated design pattern diagrams, those elementary design patterns would be split into seven *groups*. These groups are constructed based on structural or behavioral similarity among the patterns.

The next seven subsections are dedicated to explaining the key features that are common to the patterns in the same group, as well as the features that make each pattern in the groups unique. The reader is expected to be familiar to GoF's pattern. For that reason, the complete description of the pattern won't be presented in this paper, but rather the following seven subsections will only focus on the similarities and uniqueness of the patterns.

¹ GoF, or Gang of Four, is the commonly used short form of the four authors of *Design Patterns - Elements of Reusable Object-oriented Software* - number 12 in the reference list.

2.1 Plain Single Family (PSF)

This group consists of one abstraction hierarchy, and mainly only this hierarchy. The uniqueness of each pattern is explained as follows:

- i. Template Method: The *simplest* form. Other patterns in this group can be thought of as a form of template.
- ii. State: Has *aggregated* pluggable behavior.
- iii. Strategy: Has *non-aggregated* pluggable behavior.
- iv. Prototype: Strategy pattern with *pluggable cloning* method.
- v. Adaptor: Template of *hybrid* or composed classes.

2.2 Managed Single Family (MSF)

This group consists of one abstraction hierarchy, and the client gains access to it through a *middle man*.

- i. Command: The middle man is the *invoker* class.
- ii. Flyweight: The middle man is the *flyweight factory* class.

2.3 Single Intra-Family Collaboration (SIFC)

This group consists of one abstraction hierarchy with some kind of collaboration exists *within* the hierarchy.

- i. Composite: Collaboration in form of aggregate of *exclusive* copies of parents in children.
- ii. Decorator: Collaboration in form of aggregate of parents in children (*might not* be exclusive copies).
- iii. Interpreter: Collaboration in form of aggregate of parents in children, with a possible use of global *context object*.
- iv. Chain of Responsibility: Collaboration between *parents*.
- v. Proxy: Collaboration between *children*.

2.4 Two Family Collaboration (TFC)

This group consists of two collaborating class hierarchies.

- i. Bridge: Collaboration only between *parents* of the families.
- ii. Observer: Collaborations of *opposite directions* at parents and children level between families. Usually with aggregation but it is optional.
- iii. Mediator: Collaborations of opposite directions at parents and children level between families. Usually *no* aggregation.

2.5 Family Optional (FO)

This group of patterns consists of non-relating classes.

- i. Singleton: Make use of *static* property of class.
- ii. Memento: Dynamic creation of *pooled* states.

2.6 Object Builder (OB)

This group mainly serves as object creators.

- i. Builder: Involves only *one* family tree.
- ii. Factory Method: Involves only *two* family trees.
- iii. Abstract Factory: Involves *more than two* family trees.

2.7 Looping Strategy (LS)

This group mainly serves as aggregate processors.

- i. Iterator: Looping object is *provided by the aggregate*.
- ii. Visitor: Looping object is *given to the aggregate*.

3 Dedicated Design Pattern Diagram

The general construct of the proposed dedicated design pattern diagram is as follows:

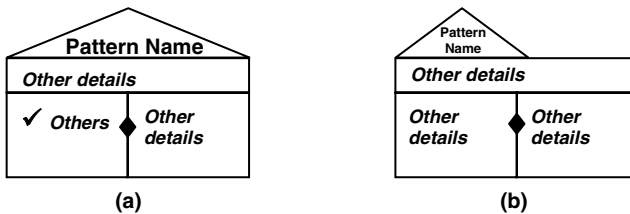


Fig. 1. Design Pattern Diagrams

Except for the triangular shape at the top, this diagram resembles class diagram. The triangle holds the pattern name with the base can be as wide as the whole box or just nice to hold the name. Unlike class diagrams, that contain the information about methods and properties of the class, design pattern diagrams contain the information about the *classes* that form the pattern. By putting only names of classes in design pattern diagrams, the cluttering caused by the details in class diagrams is greatly reduced.

There are additional infrastructure tokens to be used with the box:

- i. *Check*: Denotes that the class is provided by a framework.
- ii. *Black Diamond*: Denotes *sole mate class pairs* or couples. A sole mate pair is a pair of classes or children that are mainly collaborating only between the two of them, and less or none with other classes in the pattern.
- iii. *Connector Line*: Denotes patterns collaboration, or more specifically, denotes which class in a certain pattern collaborates with which class in another pattern. More on this will be in Subsection 3.2.
- iv. *Other relevant UML infrastructure*: Since this diagram is honoring UML, all other UML infrastructures that might fit into the diagram can be used with it. The ones that might be used mostly are the stereotype and abstract class indicator.

Note: It is important to stress that the rules that will be presented in this paper (especially in this section) would be a set of loosely defined and implemented rules. This non-stringent-rules approach is inline with the way the rules of UML and design pattern have been implemented. While honoring preciseness as the main target in [14][22][24], this paper has to deviate from this mainstream works as it has its own target, i.e. to reduce the cluttering in design diagram at inception phase. Maintaining the way of UML and GoF's design pattern is the best option to achieve this.

3.1 Conversion Rules

This subsection describes, in writing, the rules to convert the OMT or UML based GoF's design pattern into this new dedicated design pattern diagram. Please refer to Figure 8 - 9 in the Appendix for the complete list of actual visual conversion.

- i. *Orphan classes:* If the pattern contains any orphan classes (classes that are not extended from any abstract parent, like the invoker and receiver classes in command pattern), these classes have to be listed first, in any order, in the box and in single column.

Applicability: MSF patterns as they have orphan middle men; FO patterns as all classes are orphans; and the visitor pattern since the aggregate is part of the pattern.

- ii. *Parent's name:* Put parent's names before any children's names but after the orphans. If there is any orphan exists, put a horizontal double line separating them from the abstract parents. If there are only one or two families, the parent names are in one or two columns respectively. If there are more than two families, like in abstract factory pattern, everything is in one column and each family is separated by a horizontal double line.

Applicability: All patterns except adaptor.

Exception: Since *adaptor* pattern is using other classes, which might be orphans, as adaptees or for forming a hybrid, these classes need to be listed side by side with the respective children. This means two columns are needed for children-adaptee pairs, but the parent is only written on top of the children's column. The box on top of the adaptees is empty, *unless* the adaptees belong to another parent.

- iii. *Children's name:* If there are two families, the children are written side by side in two columns. Put a diamond on the border between them if they are sole mate pairs.

Applicability: All LS, TFC, and factory method pattern.

If there is only one family, and if there is no distinct two groups of children, then list the children in one column.

Applicability: Chain of responsibility, builder, and all MSF and PSF patterns except adaptor (for adaptor, see (ii) above).

If there is only one family, but there are two distinct groups of children, then list them in two columns, and pair them side by side with a diamond if they are collaborating.

Applicability: All SIFC patterns except chain of responsibility.

Exception: The abstract class *decorator* in decorator pattern is considered a child since it is a child of another abstract class *component*.

Note: In many cases, parent or abstract class may be implemented as interfaces. As a matter of fact, when two patterns collaborate, the affected classes have to inherit from more than one base or parent classes. In languages that prohibit multiple inheritance, this can only be done with interface.

3.2 Layout Scheme

The Need for Connection Network. Having a software architecture expressed in form of design patterns means there must be interconnections between the patterns. Attempts to visually highlight design patterns in a software system using existing UML infrastructures or by dotted or shaded area as in [5][7] have shown an increase in the complexity of the diagram. The dotted or shaded areas of the collaborating patterns would show *overlapping* classes. This is understood as the collaboration is achieved through the sharing of these classes. This also explains why multiple inheritance or multiple interfaces are needed for the shared classes – these classes need to inherit from more than one abstract parents (or more than one interfaces) that make up the collaborating patterns.

The conversion rules described in Subsection 3.1 reduce the complexity of the software architecture diagram by removing the lines between the class diagrams. How the classes in the respective design pattern diagram are connected are as GoF described in their book [12]. However, the relationship between the design patterns still needs to be shown. Since the collaborating patterns are sharing some common classes among them, such classes will be *repeated* in the design pattern diagram. In order to *highlight* this repetition, these common classes would be connected using a line between the two design pattern diagrams.

Each and every design pattern diagram in a system must be connected to at least one other design pattern diagram. Otherwise the pattern is not used, or the pattern is used only by the client or main program which is normally not shown in the system.

Connection Rules. The simple connection rule between the repeated classes in the collaborating design patterns as explained in the previous sub-subsection has a potential to re-introduce complexity into the design diagram. To avoid such issues, a few rules are proposed so that the connections between the design pattern diagrams can be drawn without adding any complexity to the entire diagram.

- i. *Connect only concrete classes:* Design patterns may seem to share the abstract and the concrete classes, but actually the sharing of the abstract one is understood from the structure. So it is necessary only to show the link between the concrete ones, unless the sharing actually involves the abstracts.
- ii. *Only straight line horizontally or vertically:* As much as possible, stick to drawing the connectors horizontally or vertically as straight lines only.
- iii. *Right angle bends or curves:* If it is unavoidable, bending or curving at right angle can be done. Curves should be drawn at smallest radii possible.
- iv. *Linking to a class at opposite side:* If it is unavoidable, a connector line can be drawn to join a design pattern diagram with a triangle (Figure 2) to show that it is connected to the class at the opposite side rather than the one it is joining to.

v. *Passing through a diagram*: If it is unavoidable, a connector line can be drawn passing behind a design pattern diagram with a *stopper* line at both sides (Figure 3). If there are many lines involved, the same sequence should be maintained at both sides. Optionally, the lines can be labeled with matching encircled numbers at one or both ends. They are compulsory if the sequences at the sides are different (Figure 4).

Note: Just like other UML diagrams, the above connection rules are only loosely enforced. Should it cause confusions or should it cause even more cluttering, then it is advised not to follow the above rules. Stick to whichever method that really solve the cluttering problems.

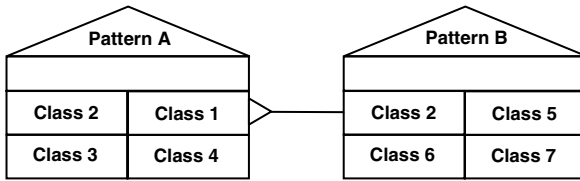


Fig. 2. Linking to opposite side - Pattern A and B share Class 2

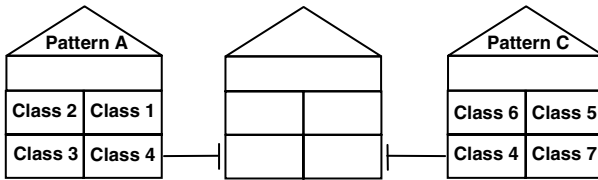


Fig. 3. Link passes behind - Pattern A and C share Class 4

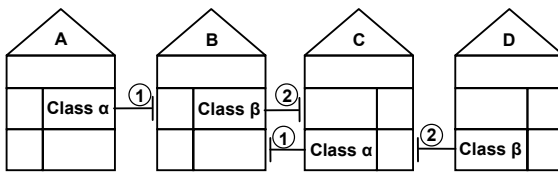


Fig. 4. Swapped links with labels - A and C share Class α ; B and D share Class β

4 Examples

4.1 Iterable Composite

A popular tutorial question in a design pattern course is to ask the students to implement iteration on a composite pattern. This can be shown as the following UML diagram and the conversion:

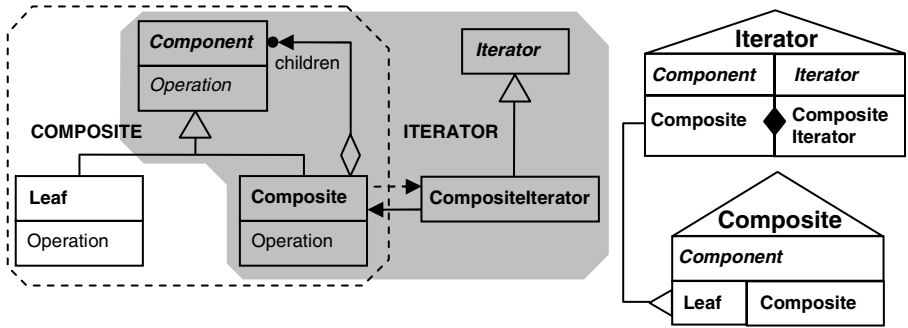


Fig. 5. Iterable Composite Pattern

4.2 Conversion of Dong's Example [5]

Figure 6 is a simplified re-drawn figure of Dong's example in [5], and the conversion of it into the proposed design pattern diagram. As opposed to Dong's example, the converted version opts to un-share the abstract Component class between the patterns Composite and Decorator – which makes the two patterns share only the class Content. This should be a better design as it increases coherence and reduces coupling. This also means the class Content needs to implement two interfaces.

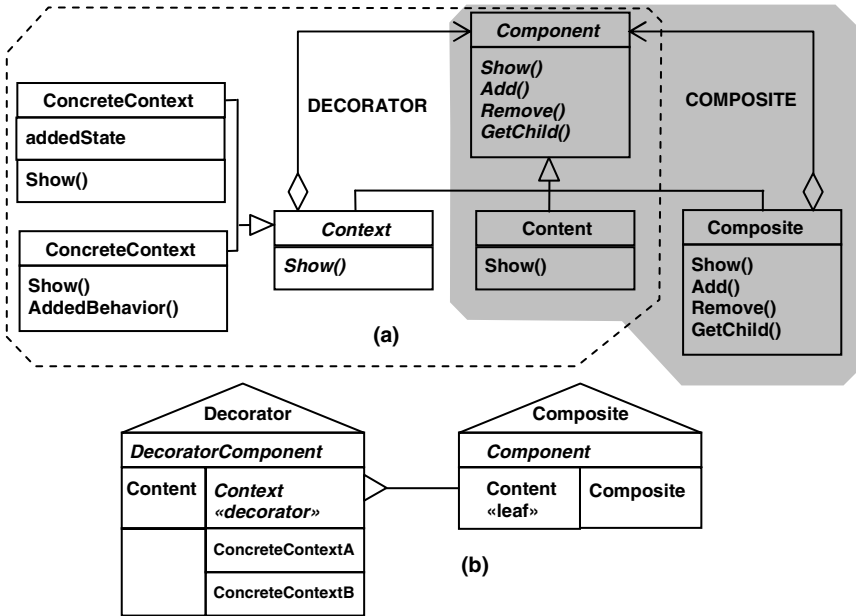


Fig. 6. Conversion of Dong's Example

4.3 Adaptor with Strategy

When two incompatible systems are connected with an adaptor pattern, it is extremely cumbersome if the adaptee system is always updated since the adaptor needs to be updated accordingly. This can be solved if the adaptee side is implemented as strategy pattern:

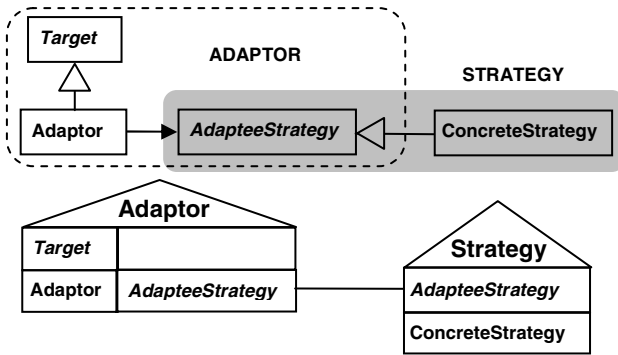


Fig. 7. Adaptor with Strategy Pattern

4.4 Singleton

Singleton is the only design pattern that is completely representable by class diagram *alone* since it is composed of only one class. See the conversion in Figure 8(d).

5 Conclusion

The most immediate and significant use of this diagram is during the inception phase [3][21]. At this stage, the developers normally would like to see the big picture of the project which makes the top-down design mode more convenient.

The patterns would first be laid out on the design diagram, and then the interconnections between them would be decided. Two patterns are interconnected if they share some common classes. These shared classes play extra roles as they are playing one role on one end of the connection (pattern) and another role on the other end. These interconnections would, in a very natural manner, give us the solution model for the problem domain by suggesting what extra roles that the classes in the respective pattern should play with respect to the pattern that it is connected to. The operation or method contracts would also be easy to be anticipated just by looking at the type of the pattern used. Once the pattern roles and those extra roles played by the classes are finalized, it is easy to start with the proper OOD mode of development, i.e. bottom-up. This is a great help in designing the system, for example, using CRC session approach [38].

Other than just a clearer layout of software architecture, the proposed design pattern offers some more advantages as follows:

- i. *Less documentation*: The diagram is self-documenting. The structure of the individual pattern is known and the relation among them is clearly pictured.
- ii. *Clear highlight of update hotspot*: Since the diagram consists of GoF patterns, the places where the update would normally be done are already documented by them.
- iii. *Clearer class hierarchy documentation*: The information about which class extends which class is obvious from the individual diagrams, and any multiple inheritance or interfaces can be seen from the connection of the classes.
- iv. *Ensures the best granularity*: Example 4.2 shows how easy it is to split (refactor) an abstract class based on which pattern it belongs to. The sharing nature shown in the diagram can effectively suggest the best granularity level for the system.

5.1 MOF Definition

There is clearly a need to define the diagram proposed in this article using MOF (meta-object facility). At this stage of the research however, it is decided that it would be good to expose to the world about the idea of having such diagram first before proceeding to the formal definition of the diagram. The move to MOF definition would definitely be in the next future step of this work.

5.2 What This Work Is NOT

Due to the numerous strong mainstream works on design patterns that are not in the same direction as this article that might mislead our readers, it is good to make some clear proclamation statements about the direction of our works. These statements are about what our works are NOT meant for, and the purpose is to avoid any *unfair* expectations from our work that are not in our research's problem statements. The author also hopes that, by stating these clear statements of disclaimer, the mind can be made more open to new directions in this field instead of following the mainstream works that might already be close to saturation.

- *This work is NOT meant to expressively define design patterns*. Due to this, the proposed diagram set won't have any feature for such purposes either in the patterns' structure, behavior, constraint or instantiation. This article honors all cited works related to patterns definition but our problem statement is different, i.e. to compactly visualize design pattern at architecture level to hide the class details, instead of visualizing at pattern level that reveals the class details.
- *This paper is NOT meant to catalogue all existing design patterns*. Even though the author is advocating patterns of GoF's, it doesn't mean the existence of other patterns is unnoticed. At this level GoF's patterns are only accepted as the most elementary patterns – due to the fact that they are among the earliest documented – and the author expects to be able to define

most of later patterns in term of GoF's patterns. The proof of this theory or otherwise would be part of the next step in this work.

- *This work is NOT, in any way, forcing the programming community to memorize all GoF's patterns.* The popularities of GoF's patterns are not all the same. An experienced programmer can tell the structure of the most widely used GoF's pattern by heart and conveniently draw the diagram of it.

References

1. Antonioli, G., Casazza, G., Di Penta, M., Fiutem, R.: Object-oriented design patterns recovery. *Journal of Systems and Software* 59(2), 181–196 (2001)
2. Buschmann, F., Henney, K., Schmidt, D.C.: *Pattern-Oriented Software Architecture: On Patterns and Pattern Languages*. John Wiley & Sons, Inc., Hoboken (2007)
3. Cantor, M.R.: *Object-oriented Project Management with UML*. John Wiley & Sons, Inc., New York (1998)
4. Dong, J.: UML Extensions for Design Pattern Compositions. *Journal of Object Technology* 1(5), 149–161 (2002)
5. Dong, J.: Representing the Applications and Compositions of Design Patterns in UML. In: *Proceedings of The 2003 ACM Symposium on Applied Computing*, Melbourne, Florida, USA (March 2003)
6. Dong, J., Yang, S.: Visualizing Design Patterns with A UML Profile. In: *Proceedings of IEEE Symposium on Human Centric Computing Languages and Environments*, Auckland, New Zealand (October 2003)
7. Dong, J., Yang, S., Zhang, K.: Visualizing Design Patterns in Their Applications and Compositions. *IEEE Transactions on Software Engineering* 33(7), 433–453 (2007)
8. Eden, A.H., Yehudai, A.: Patterns of the Agenda. In: Dannenberg, R.B., Mitchell, S. (eds.) *ECOOP 1997 Workshops*. LNCS, vol. 1357, pp. 100–104. Springer, Heidelberg (1998)
9. Eden, A.H., Yehudai, A., Gil, J.: Precise Specification and Automatic Application of Design Patterns. In: *Proceedings of 12th IEEE International Conference on Automated Software Engineering*, Incline Village, Nevada, USA (November 1997)
10. Fowler, M.: *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, 3rd edn. Pearson Education, Inc., Boston (2004)
11. France, R.B., Kim, D.K., Ghosh, S., Song, E.: A UML-Based Pattern Specification Technique. *IEEE Transactions on Software Engineering* 30(3), 193–206 (2004)
12. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: *Design Patterns - Elements of Reusable Object-oriented Software*. Addison-Wesley, Indianapolis (1995)
13. Gasparis, E., Nicholson, J., Eden, A.H.: LePUS3: An Object-Oriented Design Description Language. In: Stapleton, G., Howse, J., Lee, J. (eds.) *Diagrams 2008*. LNCS (LNAI), vol. 5223, pp. 364–367. Springer, Heidelberg (2008)
14. Le Guennec, A., Sunyé, G., Jézéquel, J.-M.: Precise Modeling of Design Patterns. In: Evans, A., Caskurlu, B., Selic, B. (eds.) *UML 2000*. LNCS, vol. 1939, pp. 482–496. Springer, Heidelberg (2000)
15. Hsueh, N.L., Chu, P.H., Chu, W.: A quantitative approach for evaluating the quality of design patterns. *The Journal of Systems and Software* 81(8), 1430–1439 (2008)

16. Huang, K.C., Wang, F.J., Tsai, J.H.: Two design patterns for data-parallel computation based on master-slave model. *Information Processing Letters* 70(4), 197–204 (1999)
17. Jakubík, J.: Modeling Systems Using Design Patterns. In: *Informatics and Information Technologies Student Research Conference*, April 2005, pp. 151–158. Slovak University of Technology, Bratislava, Slovakia (2005)
18. Kim, D.K., France, R., Ghosh, S., Song, E.: A UML-Based Metamodeling Language to Specify Design Patterns. In: *Proceedings of Workshop on Software Model Engineering (WiSME) with Unified Modeling Language Conference*, San Francisco, California, USA (2003)
19. Kim, S.K., Carrington, D.: Using Integrated Metamodeling to Define OO Design Patterns with Object-Z and UML. In: *11th Asia-Pacific Software Engineering Conference*, Busan, Korea (November 2004)
20. Korhonen, A., Malmi, L., Saikkonen, R.: Design Pattern for Algorithm Animation and Simulation. In: *Proceedings of the First Program Visualization Workshop*, Joensuu, Finland (July 2000)
21. Larman, C.: *Applying UML and Patterns: An Introduction to Object-oriented Analysis and Design and Iterative Development*, 3rd edn. Pearson Education, Inc., Upper Saddle River (2005)
22. Lauder, A., Kent, S.: Precise Visual Specification of Design Patterns. In: *12th European Conference on Object-Oriented Programming (ECOOP 1998)*, Brussels, Belgium (July 1998)
23. De Lucia, A., Deufemia, V., Gravino, C., Risi, M.: Design pattern recovery through visual language parsing and source code analysis. *Journal of Systems and Software* 82(7), 1177–1193 (2009)
24. Mak, J.K.H., Choy, C.S.T., Lun, D.P.K.: Precise Modeling of Design Patterns in UML. In: *Proceedings of 26th International Conference on Software Engineering (ICSE 2004)*, Scotland, UK (May 2004)
25. *Object Constraint Language Version 2.0* (2006), Object Management Group, <http://www.omg.org/cgi-bin/doc?formal/06-05-01.pdf>
26. *OMG Unified Modeling Language (OMG UML), Infrastructure, Version 2.2* (2009), Object Management Group, <http://www.omg.org/spec/UML/2.2/Infrastructure>
27. *OMG Unified Modeling Language (OMG UML), Superstructure, V2.1.2* (2007), Object Management Group, <http://www.omg.org/spec/UML/2.1.2/Superstructure/PDF>
28. Pasala, A., Ram, D.J.: FlexiFrag: A design pattern for flexible file sharing in distributed collaborative applications. *Journal of Systems Architecture* 44(12), 937–954 (1998)
29. Pauwels, S.L., Hübscher, C., Bargas-Avila, J.A., Opwis, K.: Building an interaction design pattern language: A case study. *Computers in Human Behavior* 26(3), 452–463 (2010)
30. Rasool, G., Philippow, I., Mäder, P.: Design pattern recovery based on annotations. *Advances in Engineering Software* 41(4), 519–526 (2010)
31. Riehle, D.: The Perfection of Informality: Tools, Templates, and Patterns. *Cutter IT Journal* 16(9), 22–26 (2003)
32. Röbbling, G.: A First Set of Design Patterns for Algorithm Animation. *Electronic Notes in Theoretical Computer Science* 224, 67–76 (2009)

33. Schleicher, A., Westfechtel, B.: Beyond Stereotyping: Metamodeling Approaches for the UML. In: Proceedings of the 34th Hawaii International Conference on System Sciences (HICSS), Maui, Hawaii, USA (January 2001)
34. Smith, J.M., Stotts, D.: Elemental Design Patterns: A Link Between Architecture and Object Semantics. Technical Report. Department of Computer Science, University of North Carolina at Chapel Hill, North Carolina, USA (2002)
35. D'Souza, D., Auletta, V., Birchenough, A.: First-Class Extensibility for UML - Packaging of Profiles, Stereotypes, Patterns. In: France, R.B. (ed.) UML 1999. LNCS, vol. 1723, pp. 265–277. Springer, Heidelberg (1999)
36. Sunyé, G., Le Guennec, A., Jézéquel, J.-M.: Design patterns application in UML. In: Hwang, J. (ed.) ECOOP 2000. LNCS, vol. 1850, p. 44. Springer, Heidelberg (2000)
37. Wernhart, H., Kühn, E., Trausmuth, G.: The replicator coordination design pattern. *Future Generation Computer Systems* 16(6), 693–703 (2000)
38. Wilkinson, N.M.: Using CRC Cards: An Informal Approach to Object-oriented Development, SIGS Books, New York (1995)
39. Yacoub, S.M., Ammar, H.H.: Pattern-Oriented Analysis and Design. Addison-Wesley, Boston (2004)
40. Yacoub, S.M., Ammar, H.H.: Pattern-Oriented Analysis and Design (POAD): A Structural Composition Approach to Glue Design Patterns. In: Proceedings of 34th International Conference on Technology of Object-Oriented Languages and Systems (TOOLS), Santa Barbara, California, USA (July 2000)
41. Zdun, U.: Some patterns of component and language integration. In: Proceedings of EuroPLOP 2004, Irsee, Germany (July 2004)
42. Zdun, U., Avgeriou, P.: Modeling Architectural Patterns Using Architectural Primitives. In: Proceedings of the 20th annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, San Diego, California, USA (October 2005)
43. Zhu, H., Bayley, I., Shan, L., Amphlett, R.: Tool Support for Design Pattern Recognition at Model Level. In: 2009 33rd Annual IEEE International Computer Software and Applications Conference (COMPSAC 2009), Seattle, Washington, USA (July 2009)

Appendix: The Complete Catalogue of GoF's Design Pattern Conversion

This appendix (Figure 8 and 9) presents the complete list of GoF design patterns with their conversion into the new design pattern diagrams. This list can be used as reference in converting a bigger system as long as it is based on GoF's patterns.

Should there be a discovery of a genuinely new pattern, for the convenience of everyone, it is suggested that the discoverer himself should document the corresponding conversion into this new diagram after considering all criteria proposed in this paper. The author of this paper is not in any way tries to force a consensus in the conversion of any new pattern – the discoverer's decision should be honored.

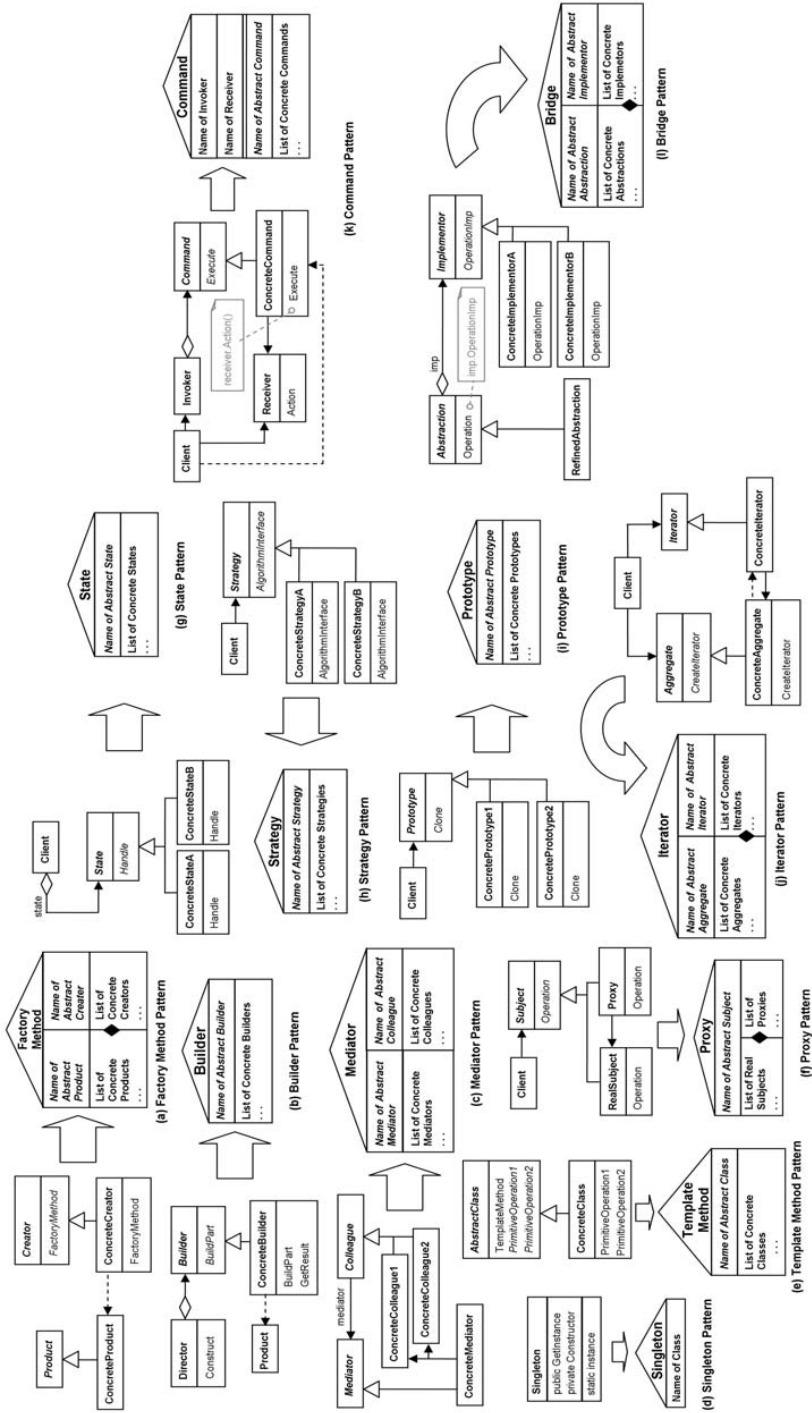


Fig. 8. Complete Conversion Catalogue (Part I)

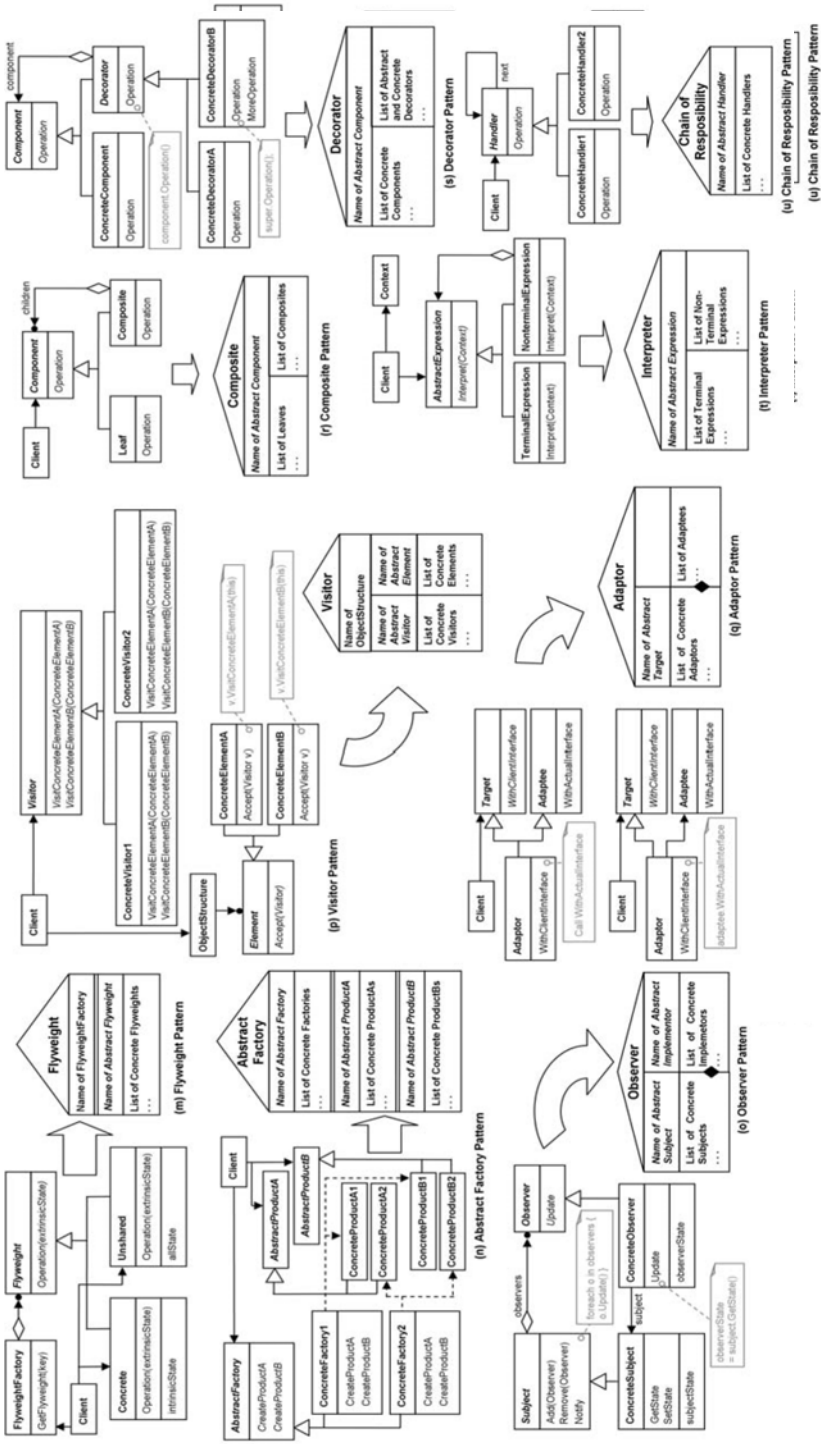


Fig. 9. Complete Conversion Catalogue (Part II)

Towards Natural Interaction with Wheelchair Using Nintendo Wiimote Controller

Mahmood Ashraf and Masitah Ghazali

Department of Software Engineering,
Faculty of Computer Science and Information Systems,
Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia
amahmood4@live.utm.my, masitah@utm.my

Abstract. The importance of natural interaction increases when the subjects are disabled people. It has been found that embedded solutions for impaired people lack in fluid interaction properties. This work investigates the interface design of a wheelchair controller according to the rules of physicality. Our aim is to provide further ease of use to the impaired whilst strengthening link between embedded software engineering and human-computer interaction. We propose an improved multi-function interface design by using Nintendo's Wiimote that is more intuitive, robust, low cost, and most importantly, natural to use.

Keywords: Physicality; embedded software; wheelchair interface design; usability; natural interaction.

1 Introduction

Interaction plays pivotal role in our daily life. We spend most of the time in different kinds of interaction. Some artefacts are felt easier to use than others. Do the devices around us map their physical appearance and working with their logical functionalities? The stronger this mapping, the more natural interaction will take place. The hidden reason behind our liking of some devices as compared to their counterparts is due to these strong mappings. In addition, there exist many other aspects of an object that result in natural interaction.

People having physical disabilities are more affected by the quality of interaction. Impaired people have lesser choices and hence are less flexible than normal ones. This deficiency must be covered by the devices they use. Therefore, the devices like wheelchair used by the impaired people should be more carefully designed. A wheelchair can have different kinds of interfaces based on its specific type, that is, manual-driven; power-driven; or intelligently-power-driven. The natural interaction is the combination of human innate abilities and the physical visceral qualities in the artefacts [1]. The visceral quality is that physical aspect of device which recruits our natural human abilities [2]. Physicality as defined by Donald A. Norman [3] is "the return to mechanical controls, coupled with intelligent, embedded processors and communication". The importance of physicality is evident from many examples. The notational and social aspects of physical artefacts cannot be ignored in agile software

development [4]. Pilots of commercial airlines use papers for many purposes including managing attention [5]. Physical representations are difficult to ignore than digital reminders [6]. McKinnon [7] states, “(Some software) helps you effortlessly create new ideas, break them down, arrange them, colour code them but most importantly - print them out to use them as technology in their own right”.

Nintendo’s Wii is presently the largest selling video game around the world. The controller of the game is called Wii remote or Wiimote. The reason behind using Nintendo’s Wiimote for wheelchair interface is its better interaction capabilities. It has more flexibility and scalability than conventional components used in wheelchair interface. Wiimote has a quad-directional button also called control pad, eight other buttons, 3D motion sensing, and pointing functions for input purpose and a speaker, rumble pack, and four LEDs for sound, tactile, and visual feedback, respectively.

This paper is organized as follows. The next section describes motivation and related work followed by introduction of design principles and then existing system. Next, proposed method is discussed. We then compare the physical and logical mappings. Before discussion and conclusion we analyse our proposed design in the light of design principles.

2 Motivation and Related Work

Among the vast application areas of embedded software systems, we look into the interface design of assistance devices for physically impaired people. Although a lot of work has been done in the past on the solutions for the impaired people but the specific area of interface design has not yet received appropriate attention. In order to have a deep understanding of the requirements-availabilities relationship, we investigate the problems and solutions of firstly blind impaired people, and secondly the mobility impaired people. In this way we can be better able to design a natural interface for wheelchair users keeping in mind that blind users also need to interact with the wheelchair.

For our study, we also visited hospitals, impaired people’s care centers, interviewed impaired children and their caregiver staff, and observed the ways in which impaired people interact with their wheelchairs to gain first-hand knowledge of their needs. The impairment that leads to the use of wheelchair may be caused by many diseases or injuries. Some of the common problems are listed in Table 1.

Table 1. Problems that may result in need of a wheelchair

Category	Name of Disease
Spinal	Degenerative Disc, Facet Arthritis, Herniated Nucleus Pulposus, Osteoarthritis Scoliosis, Spina Bifida, Spinal Stenosis, Spinal Arachnoiditis, Vertebral Fracture
Muscular	Hypotonic Quadriplegia, Juvenile Rheumatoid Arthritis, Paralysis, Muscular Dystrophy
Brain	Cerebral Palsy, Epilepsy, Guillain-Barré Syndrome, Parkinson
Legs and feet	Bowlegs, Knock-knees, Pigeon Toes
Injuries/By birth	Damage or absence of any mobility linked organ

Recently some efforts have been made to bridge the gap between software engineering and human-computer interaction and to provide ease to the embedded software developers. Kim et al. [25] have proposed a user behavioral analysis framework for the ubiquitous embedded systems. Bujnowski et al. [8] empirically analyzed the use of tactile system to guide the visually impaired people during walk. Although they used tactile vibrators on subject's one arm only, still their results showed that tactile feedback is more comprehensible to the blind. This work can be enhanced easily by increasing the directions from three to five or even more. At each arm, small duration vibration would mean turn left 45°, and long vibration would mean turn left 90°, while vibration on each arm simultaneously would mean to move forward. The study by Hara et al. [9] has also proved that the tactile feedback is better than audio, especially in outdoor's possibly noisy environment. The results of Shah et al. [10] have also confirmed the former studies. Based on these studies among many others it can be concluded that the tactile sense of visually impaired people is more sensitive and better than audio feedback.

Ivanchenko et al. [11] have proposed a computer vision based solution. A camera and high speed computing device for graphical processing of images made the system costly besides other flaws. This approach targets the visually impaired users who may have additional mobility difficulties. This system engages an arm of user all the time which is laborious especially for an impaired person. It is difficult for fixed camera to monitor the free moving cane. Lastly, the computer vision program needs improvement by categorizing the friends and foes among obstacles. Kuno et al. [12] have come up with even costlier wheelchair interface solution having multiple cameras, high speed computing machines for image processing, and automated control of wheelchair. The solution has overwhelmed the user with many controlling points and strict limitations on head movement for the user. Any slight movement of head for communication with some person or for enjoying the environment will result in the unintentional change of direction of wheelchair that may end up in an accident. The system is designed in a way that the back camera tracks and follows the movement of caregiver. However, the back camera can interpret any pedestrian as caregiver because authors have not designed anything to identify the caregiver. As the caregiver control has priority over user control, in case of wrong selection of caregiver the user is helpless especially at a busy place like market. This system indicates a lot of enhancements to be made on the interface of the system besides functionality. Abascal et al. [13] have proposed a mobile interface for the patients of quadriplegia (who are unable to use their arms and legs) that is low cost, automatic, and requires less effort by the user. It also takes into account the activeness of user for rehabilitation purposes. The user can select the available paths after scanning a matrix of icons, with a pushbutton or a joystick. To select a destination the user is provided with a hierarchical map model due to compact menu-based display. The presented entries for the destination to the user are optimized by two ways. First, only the reachable destinations from the current point are displayed to reduce the time and effort of user in selection. Second, the displayed options are ordered based on the frequency of selection by the user. However, the user interface needs enhancement. In all the discussed scenarios we have found spaces for improvement in the user interface.

3 Introducing the Design Principles

The design principles and physicality rules have not yet applied on the wheelchair interface to introduce natural interaction. Embedded software developers emphasized on functionality by providing multiple complex interfaces simultaneously for a single chair whilst ignoring the usability and fluid interaction aspects, completely. Users like a naturally used device although how simple it is but dislike a very sophisticated device having poor interaction. We briefly discuss here the design principles for natural interaction. We will evaluate the existing system and compare with proposed system according to these principles in detail in section 7.

If a control expresses its underlying logical state by its physical state then this control holds the property of exposed state. For example, simple on-off light switches. If the physical appearance does not express the logical state then it is called hidden state. For example, twist control of a speaker. The directness of effect property is directly proportional to the action performed. A small push results in small movement and a large push results in large movement. Locality of effect means the result of an action should be there and then. A control having bounce back effect maintains a state until operated then either stays or returns back to its initial physical state. For example, push button. Cultural influence indicates the frequency of usage in a society. Affordance is the number of action options perceived by the user. Compliant interaction shows the symmetrical aspect of interaction between user and system. Physical and mental requirements are the amount of physical and cognitive efforts, respectively that are needed to perform an operation while interacting with a control.

4 Existing System

A wheelchair is of many types ranging from manual house-hold to sporty powerchair, and from hand-operated to mind-operated. The core functions a wheelchair must provide are listed in Table 2.

Table 2. Functions a wheelchair must provide

Category	Functions
Seating	Seating space for passenger Back support Arm rest Foot rest
Movement and Control	Move forward Turn right Turn left Stop Move back Rim for self control
External/Caregiver Support	Handles Push Pull

The manual wheelchair does not offer power, related controls and accessories. No matter how sophisticated or advanced a wheelchair is, it must provide the functions listed in Table 2. Wheelchair can be broadly categorized as follows:

1. Manual control using rim
2. Power control using guidance devices
3. Power control with intelligence using sensors and/or cameras

The existing system [15] is an example of embedded system, a wheelchair having triple-interface that can be operated by power or manually by using rim (Fig. 1).

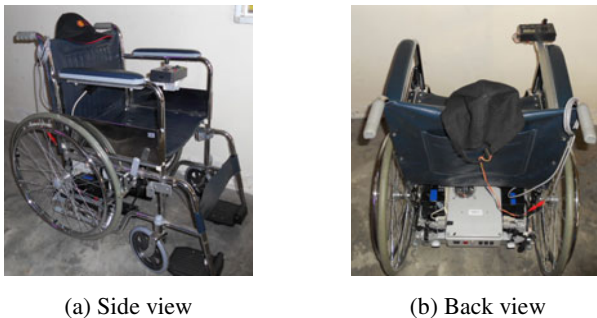


Fig. 1. Existing system – a wheelchair with three user interfaces [15]

This wheelchair does provide the core functionalities that any wheelchair must provide (Table 2). Additionally, it has features of obstacle detection, collision avoidance, maneuvering with the help of sensors at the front of the rider's cap. The wheelchair takes command with the movement of head (having cap). If rider turns head right, the wheelchair moves right and same for other three directions.

Therefore, the three interfaces are; head-control with power, hand-control with power, and manual hand-control through rim. For the scope of this paper, we are targeting the second interface that is hand-control with power. The interface consists of a controller box mounted on the right arm rest of the wheelchair consisting of a joystick, two buttons, and four LEDs as shown in Fig. 2. For our study, we will not discuss the functionality of each (only design) except the joystick (both design and functionality) assuming these are detached from their underlying functionality.

4.1 The Controller Box

The controller box is mounted on a square metal bar that is attached besides the right arm rest of the wheelchair perpendicular to it. It hinders in the way of user whilst seating or leaving the wheelchair. The size of the controller box is also large (width 12.5cm x length 7cm x 3cm thickness). In Fig. 2 (b), a strip with a hole in its center is visible on the right side of the box. This strip is also present on the left side. This is speaking by itself that this box was not meant to be placed or mounted here. The holes are for screws. A separate metal bar is used to host the box instead of the arm of wheelchair. The width of additional metal rod is 2.4cm that further reduces the seating space of user.



(a) Top view

(b) Side view

Fig. 2. Existing wheelchair controller mounted on a bar besides the right arm of the wheelchair

4.2 The Joystick

The joystick is occupying further 3cm height resulting in a total height of 6cm from the wheelchair arm. Furthermore, there is no labeling for guidance to indicate any direction. Among 360°, user cannot predict the operational range of joystick. This is an example of lack of affordance. The joystick is a bounce back control [16] as it returns to its initial position after the user releases the pressure. A good bounce back control should have good affordance. But this control has a hidden state property and therefore it needs to have some labeling for the directions [17].

In addition, this joystick needs to be grabbed or grasped with fingers to operate. Whilst we are focusing on the impaired users, among them patients having no fingers may also use this wheelchair, for example leprosy patients. Therefore, this control is approximately unusable or very difficult for the people having no fingers. Another problem of the joystick is that it is twistable and rotatable in clockwise and anticlockwise direction, having no logical functionality. The physical-logical mapping is absent here that will only confuse the user. One more limitation of this joystick interface is the introduction of four screws that are holding joystick module inside the box. The screws on the box will hinder the use of joystick because these are well above the surface. The edges of the screws may injure the user in any unintentional or careless handling to the device.

4.3 The Buttons

Fig. 2 (a) shows the buttons are in opposite direction to each other. This is against the consistency and will result in confusion to the user. However, each button consists of two symbols; one circle and one line. It means that when both buttons are pressed on circle, one will be up and second will be down. Another problem is that the line symbol on each button is also interestingly in opposite direction. Both lines are perpendicular to each other only resulting in more confusion to the user. The problems do not end here. Another difference is the size of buttons. Left is slightly bigger than right without any justification. The buttons are not aligned to each other; neither center, upward, or downward. The buttons are hard to press. Even a normal person has to force-press the buttons. A device for the impaired user; especially who may be impaired due to fingers or hands too; should be easy to operate. Elderly people may also be the subjects. So, soft-press buttons are recommended.

4.4 LEDs

There are four LEDs in total on the upper surface of the box. One is in front of the joystick, and the rest are in front of the two buttons. Although, they are not grouped with the joystick and buttons by some decoration or drawing on the box but still they look grouped somehow by their locality. Another problem is the colour selection. Red LED is present in both the groups that may cause cognitive problem for the user. The LEDs are also not labeled like the other controls on the box.

5 Proposed System Using Wiimote

The introduction of microtechnology followed by nanotechnology has really changed the way things look and work and it has its advantages too. The UIs should also adopt its good qualities. We no more have the mainframes and large punch cards. Everyone is looking for a compact, handy, usable, and low-cost solution. One example of this can be seen by looking in the history of computer games starting from 1940s on mainframes. Since then, not only the size of game consoles has reduced considerably, the game controllers have also become more compact and user friendly. Game controllers are the interaction devices which are being heavily used by masses. Feedback from the users in the past many decades had been helpful in maturing the design of game controllers. Although, joystick was already replaced with a push button by many companies but Nintendo's controllers have been applauded more than others. Nintendo Wii has recently introduced new design for its game controller as shown in Fig. 3.



(a) Wii console with Wiimote



(b) Wiimote engaged in hand

Fig. 3. Nintendo's Wii video game [20]

This game has broken the previous records of sales [18] and currently it is at top position (more than 11,450,000 pieces sold till December 31, 2010 [19]) among its traditional rivals; Sony's Xbox 360, and PlayStation 3. Two among other reasons for this huge success are its user friendly 3D flexible controllers, and the low-cost. These controllers have been appreciated by a large number of users.

Game controllers have been exercised in the areas other than computer games for example as an assistive device for impaired people using switch controlled software [21], limb action detector [22], 3D unistroke gesture recognition [23], and controlling robots [24]. All of these studies have used Nintendo's controllers by stating its

benefits as low cost, easy handling, and availability. We want to include two more benefits. First, instead of using different controllers for different things; resulting in handling burden, space occupation, increased memory requirements by user, time consuming, and difficulty in context switching; least number of controllers should be used (only one for all artefacts as a perfect case). Second, the studies mentioned in literature review section of the paper have shown that the user needs time to get comfortable with any interactive device. Using the expertise of game players, especially if they are impaired too, would be a good idea.

After analyzing the problems and limitations of existing method we have suggested to replace the joystick with the Wiimote controller’s quad-directional button shown in Fig. 4 (b).

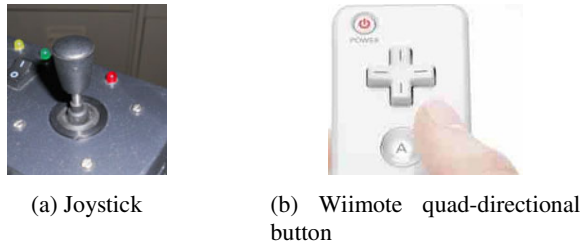


Fig. 4. Existing (a) and proposed (b) controls

The joystick can move/rotate 360°. Assuming that the forward direction corresponds to north (according to Fig. 5), the rest of the directions correspond accordingly.

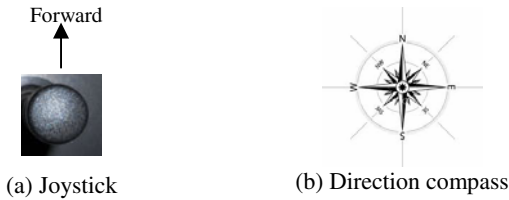


Fig. 5. Joystick directions

The mappings between joystick actions and the wheelchair functions are listed in Table 3.

The rider of the wheelchair can move forward, turn right, turn left, and move back by interacting with (pushing/pulling) the joystick. On leaving the joystick at the center (default position), the wheelchair stops. Table 3 is also listing movements at four angles and twist that have no functionality associated with them. In fact, none other than first five listed positions, there exists a physical-logical mapping. Whilst the user can move the joystick in 356 other angles without any function will only result in poor interaction. Similarly, the mappings between quad-directional button and the

wheelchair are listed in Table 4. All actions are mapped to their corresponding logical states. In addition to first five primary motions, we have proposed a double press function that is mapped to continuous forward movement of wheelchair to gain a hands-free experience. The user can press any button or the same forward button to stop the automatic forward movement. This is very important as the design is for impaired people. In existing system they have to keep holding the joystick in pushed forward position. In case of long distance, users may get tired especially due to impairment. Additionally, this also frees the user arm or hand.

Table 3. Joystick functions with their mappings

Action of Joystick	Direction	Angle	Mapped Function of Wheelchair
Push forward	North	90°	Move forward
Push right	East	0°	Turn right
Push left	West	180°	Turn left
Pull towards rider	South	270°	Move back
Stationary/Unengaged	N/A	N/A	Stop
Push forward with right	North-East	45°	Nil
Push forward with left	North-West	135°	Nil
Pull back with left	South-West	225°	Nil
Pull back with right	South-East	315°	Nil
Twist	Clock wise and anti-clock wise	Around 360°	Nil

Table 4. Quad-directional button functions with their mappings

Action of Quad-directional Button	Mapped Function of Wheelchair
Press right button	Turn right
Press left button	Turn left
Press backward button	Move back
Unengaged	Stop
Double press forward button	Move forward until any other button press

This will also facilitate those users having deformed fingers or who may not feel comfortable holding any control for long.

6 Comparison of Physical-Logical Mappings

In contrast to the joystick the quad-directional button has strong affordances. Firstly, the button is already in the shape of four directions and the user will understand its functionality by merely looking at the control before touching it. Secondly, the indicator lines on each side of the button are augmenting the affordance. Third aspect with respect to affordance is the concave shape of the button. The four edges have inclined height as compared to the center of the button. This results in intuitive

interaction with the button. Moreover, as we are considering people with any type of impairment especially mobility problems, our subjects may include people having problems of hands or fingers for example the leprosy patients. Therefore, our method requires not more than one finger to operate. Rather, people without any finger can also operate by using any edge of hand, or arm. Proposed method does not require holding or grasping anything. Moreover, proposed method does not occupy vertical space that may result in hindering with the clothes of rider, and chances of being damaged. There is no need of any separate rod to hold the Wiimote due to its smart size. It can be mounted on the arm of wheelchair easily. Current position of joystick box in the existing system is perpendicular to the arm resulting in hurdle while seating or leaving the wheelchair. Our control has been placed parallel to the arm for swift operation. There are no screws on new control in contrast to the joystick box. New system has more functional up gradation capacity and flexibility than joystick. The quad-directional button can be programmed by incorporating different commands for single, double and triple press resulting in elimination of separate button requirement. Table 5 summarizes the comparison between the existing control and the proposed method.

Table 5. Comparison between joystick and quad-directional button according to features

#	Features	Joystick	Quad-directional button
1	Mode	Input only	Input and output
2	Input operational requirement	> one finger	<= one finger
3	Output operational requirement	Not Supported	<= one finger
4	Engagement method	Hold/Grab	Press
5	Cost	High	Low
6	Vertical space occupied	3cm	0.1cm (negligible)
7	Overall space requirement	More	Less
8	Possibility to stuck with things like clothes	High	Nil
9	Probability of damage of control due to above	High	Nil
10	Up gradation/Flexibility	Nil	High

Joystick was limited to input only but the proposed system can also be used for output/feedback besides input. With the tactile feedback this interface can work efficiently for the solutions of blind users as well. We have plans to incorporate tactile feedback to our design in future.

7 Analysing the Design Principles

It is essential to investigate the design principles to achieve natural interaction. The common point in various definitions of affordance is that it invites the user to a particular action [17]. However, there are some other aspects, besides affordance, that play significant role in conveying the information about the logical function of the device to the user. The importance of these aspects or states in the design had already

proven when we explored the concept of fluidity [26] by investigating the physical and logical relationships [17]. As mentioned in section 3, due to space limitations we are briefly discussing only more influential factors in the comparison as follows:

7.1 Exposed State

The reflection of logical state through physical state is exposed state property. Due to the immediate feedback, the user easily comprehends how to manipulate the control which results in the natural interaction. In quad-direction button, the un-pressed and lifted state of button exposes the off condition, whilst the four sidedness (even in off state) exposes that it is meant to control movement in four directions. A control having exposed state does not require any additional features like markings etc, but quad-directional button has line markings on all four sides for direction indication that further augments the exposed state. Additionally, the concave shape of the button at the center and inclined four edges offer strong affordance for the finger. The joystick on the other hand, is missing the exposed state.

7.2 Hidden State

The opposite of exposed state is hidden state. Controls having this property lack naturalness. In this case additional decoration is necessary to help the user. In existing system, the joystick bears the property of hidden state but there is no decoration present. A direction indicator around the joystick would be appreciated. User does not know what the forward push will result; either it will move the wheelchair forward or backward. Some joysticks like the controller of airplane, automatic gear lever of car and controller of caterpillar increase the speed in forward direction by pulling them backwards. Joystick's strong cultural influence property at this time, however, helps the user to figure out how to move in, at least, right and left directions.

7.3 Directness of Effect

Effect is directly proportional to the action performed. In our case, both joystick and quad-directional button have fixed logical states; either on or off but joystick provides more physical movement than the quad-directional button resulting in confusion due to the property of directness of effect. Joystick user may push more than desired but the wheelchair will move at constant speed. Quad-directional button does not offer much physical movement hence user gets the feel of go or selection right after the press. This limitation will result in more natural interaction.

7.4 Locality of Effect

The result of an action should be there and then. But joystick as compared to quad-directional button has more physical movement domain. It takes considerable time, when user starts pushing or pulling from the initial position to the last physical limit, to establish the mapping between both states. Approximately close to the physical limit a meager sound of 'tick' is produced that represents that 'now' the connection has established and the wheelchair moves. Such behaviour is absent in

quad-directional button representing strong locality of effect. This is one of the reasons behind the fact that double-click or double-press action is more natural with quad-directional button than with a joystick.

7.5 Bounce Back

A control having bounce back effect maintains a state until operated then either stays or returns back to its initial physical state. For example, push button. The quad-directional button supports stronger bounce back property than wheelchair's joystick. Bounce back has two clear states; pressed or in, and un-pressed or out. Pressed state is a transient state; it only stays in this state until a force like finger is pressing it. At this moment our body becomes part of the interaction that is also called embodiment. As soon as the pressure is released, it bounces back to the out state. The affected factors within bounce back are; when the control is pressed; how long it remain pressed; when the state transition occurred physically as well as logically; are the transitions – physical and logical – mapped with each other; how much time it took in transient state after pressure is released; will it continue performing during this time too; how much time it takes to return back to its initial state; does the returning stroke has some functionality attached to it too; at the end of one press after how much time it is ready to be pressed again to successfully perform the same functionality one more time? Appropriate use of bounce back effect results in natural interaction otherwise produces confusion, as in the case of joystick of wheelchair.

7.6 Physical Requirement

Physical requirement is the amount of physical effort that is needed to perform an operation while interacting with a control or device. We need to remove extra effort and/or movement if the goal can be achieved with less effort according to the requirements. For example, a gear lever or a long joystick like control is used to change gear of an automobile. In manual transmission, five forward gears means driver has to interact with the gear lever more often. The automatic transmission cars are not bound to frequent gear shifting. Therefore, it is wise enough to replace a big control like long gear lever, requiring more effort and movement, with a smaller control. Many car manufacturers have addressed this problem, for example the latest Jaguar XKR, model 2010 has replaced the gear lever with a dial. During normal forward driving there is seldom requirement to use this control. The benefits of using such control instead of conventional large levers are multifold; reducing the extra effort, force and movement, saving the work-space for user (that is especially important inside a car having all other compact controls), reducing the chances of damage, increasing the life of control and may provide support for additional feedback (LEDs in Jaguar's example). This is the reason why a steering wheel is not appreciated on an impaired person's wheelchair. What is the need of putting extra effort if the task can be accomplished with lesser effort? Moreover, we are proposing the solution for impaired, where the requirement is to put minimum effort for achieving maximum results. Therefore, a quad-directional button is better than a joystick.

7.7 Mental Requirement

Exposed state falls into lower sub-conscious category whilst the hidden state falls into low-level cognition category of mental requirements. The tasks under sub-conscious category are natural to undertake with no burden of cognition and hence intuitive whilst low-level cognition requires some mental processing and storage. Therefore, the quad-directional button requires least cognition. The complete comparison is outlined in Table 6 with short description of design principles.

Table 6. Comparison between joystick and quad-directional button according to design principles

#	Design Principle	Description↓	Controls→	Joystick	Quad-directional button
1	Exposed state	Visible and direct mapping between physical-logical states		✗	✓
2	Hidden state	The absence of exposed state		✓	✗
3	Directness of effect	The effect is directly proportional to the action performed		Creates confusion	No confusion
4	Locality of effect	The result of an action is within temporal and spatial locality		Weak	Strong
5	Controlled state	Limitation imposed by the devices preventing user to return physical state to original position		✗	✗
6	Tangible transition	The emphasis which is given to enhance the change of states		✓	✓
7	Bounce back	Physical state remains unchanged, or return to its original position over time despite the change in logical state		Weak	Strong
8	Inverse action	Inverse logical effects being exploited by physical opposite states		✓	✓
9	Compliant interaction	Shows the symmetrical aspect of user–system interaction		✗	✗
10	Affordance	Action possibilities that are readily perceivable by a user		Weak	Strong
11	Cultural influence	How often/commonly used in a society		Strong	Strong
12	Mental requirement	Amount of mental activity (processing and memory) needed		Low-level cognitive	Sub-conscious
13	Physical requirement	Amount of physical activity (force and movement) needed		High	Low

Both joystick and quad-directional button have strong cultural influences, and share the aspects of tangible transition and inverse action.

8 Discussion and Conclusion

We have investigated the solutions for the traveling of impaired people. We also visited hospitals, impaired people's care centers, interviewed impaired children and their caregiver staff, and observed the interaction of impaired people with their wheelchairs to gain first-hand knowledge of their needs. We have found that the interaction with the interface of wheelchair controller has many serious problems. It can be improved considerably by applying the physicality rules, incorporating design principles, and improving the physical-logical mappings which has not been done before on the wheelchair controller using Wiimote. In addition to removing shortcomings of existing system, we have also added new functionalities like hands-free movement to make the life of impaired people not just easy but enjoyable.

In this work, we have investigated the existing problems in a wheelchair interface design and proposed a novel natural interface design for the impaired wheelchair users. It has many advantages over existing system including low cost, dual-mode availability (input & output), least operational requirement for input and output, easy operational method, flexibility, minimum space requirements especially vertical, no risk of sticking with clothes during operation, and no risk of damage. We have tried to provide some guidelines for developers especially embedded software engineers to help them in considering usability and naturalness of interaction during development. We are planning to further study these controls and enhance the interface design by performing comprehensive usability evaluation on impaired people.

References

1. Dix, A., Ghazali, M., Ramduny-Ellis, D.: Modelling Devices for Natural Interaction. *Electronic Notes in Theoretical Computer Science* 23–40 (2008)
2. Ghazali, M., Dix, A.: Visceral Interaction. In: *Proceeding of The British Computer Society BCS-HCL*, pp. 68–72. Edinburgh, Scotland (2005)
3. Norman, D.: The next UI breakthrough, part 2: physicality. *Interactions* 14(4), 46–47 (2007)
4. Sharp, H., Robinson, H., Petre, M.: The role of physical artefacts in agile software development: Two complementary perspectives. *Interacting with Computers* 21(1-2), (Special issue: Enactive Interfaces) 108–116 (2009)
5. Nomura, S., Hutchins, E., Holder, B.E.: The uses of paper in commercial airline flight operations. In: *CSCW 2006*, pp. 249–258 (2006)
6. Sellen, A.J., Harper, R.H.R.: *The Myth of the Paperless Office*. The MIT Press, Cambridge (2003)
7. IterEx., <http://www.planningcards.com/site/>
8. Bujnowski, A., Drozd, M., Kowalik, R., Wtorek, J.: A tactile system for informing the blind on direction of a walk. In: *Conference on Human System Interactions*, pp. 893–897 (2008)
9. Hara, M., Shokur, S., Yamamoto, A., Higuchi, T., Gassert, R., Bleuler, H.: Virtual environment to evaluate multimodal feedback strategies for augmented navigation of the visually impaired. In: *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 975–978 (2010)
10. Shah, C., Bouzit, M., Youssef, M., Vasquez, L.: Evaluation of RU-Netra - Tactile Feedback Navigation System For The Visually Impaired. In: *International Workshop on Virtual Rehabilitation*, pp. 72–77 (2006)

11. Ivanchenko, V., Coughlan, J., Gerrey, W., Shen, H.: Computer vision-based clear path guidance for blind wheelchair users. In: Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility (Assets 2008), pp. 291–292. ACM, New York (2008)
12. Kuno, Y., Yoshimura, T., Mitani, M., Nakamura, A.: Robotic wheelchair looking at all people with multiple sensors. In: Proceedings of IEEE International Conference Multisensor Fusion and Integration for Intelligent Systems, pp. 341–346 (2003)
13. Abascal, J., Cagigas, D., Garay, N., Gardeazabal, L.: Mobile Interface for a Smart Wheelchair. In: Paternó, F. (ed.) Mobile HCI 2002. LNCS, vol. 2411, pp. 373–377. Springer, Heidelberg (2002)
14. Home Care Specialists Inc.,
http://www.homecarespecialistsinc.com/medical_equipment.html
15. Sabil, S., Jawawi, D.N.A.: MARMOT and PECOS Hybrid Approach for Embedded Real Time Software Development. In: The 5th International Conference on Information & Communication Technology and Systems (ICTS 2009), Surabaya, Indonesia (2009)
16. Dix, A., Ghazali, M., Gill, S., Hare, J., Ramduny-Ellis, D.: Physigrams: Modelling Devices for Natural Interaction. *Formal Aspects of Computing* 21(6), 613 (2009)
17. Ghazali, M.: Discovering Natural Interaction of Physical Qualities to Design Fluid Interaction for Novel Devices. Research Monograph, Universiti Teknologi Malaysia (2007)
18. Ebay,
http://reviews.ebay.com/A-Comparison-of-Xbox-360-PlayStation-3-Nintendo-Wii_W0QQugidZ10000000003580375
19. Tech Watch,
<http://www.techwatch.co.uk/2010/10/04/one-in-three-uk-households-own-a-nintendo-wii/>
20. Game Spot Asia,
http://asia.gamespot.com/users/MushroomWig/show_blog_entry.php?topic_id=m-100-25803446
21. Standen, P.J., Camm, C., Battersby, S., Brown, D.J., Harrison, M.: An evaluation of the Wii Nunchuk as an alternative assistive device for people with intellectual and physical disabilities using switch controlled software. *Computers & Education* 56(1), 2–10 (2011), *Serious Games*
22. Shih, C., Chang, M., Shih, C.: A limb action detector enabling people with multiple disabilities to control environmental stimulation through limb action with a Nintendo Wii Remote Controller. *Research in Developmental Disabilities* 31(5), 1047–1053 (2010)
23. Raza, S.A., Ahmed, M.W., Madni, T.M., Tahir, M., Khan, M.I., Ashraf, M.: Preliminary evaluation of 3D unistroke gestures: An accelerometer-based approach. In: IEEE ICIIT, vol. 1, pp. 634–638. University of Central Punjab, Lahore, Pakistan (2010)
24. Olufs, S., Vincze, M.: Simple inexpensive interface for robots using the Nintendo Wii controller. In: Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems (IROS 2009), pp. 473–479. IEEE Press, Piscataway (2009)
25. Kim, W.Y., Son, H.S., Kim, R.Y.C., Jeon, B.K.: User Behavior Analysis Framework (UBAF): Mapping HCI with SE. In: Future Generation Communication and Networking (FGCN 2007), vol. 2, pp. 565–568. IEEE Computer Society Press, Washington DC, USA (2007)
26. Dix, A., Finlay, J., Abowd, G., Beale, R.: *Human-Computer Interaction*, 3rd edn. Prentice-Hall, Englewood Cliffs (2004)

Meta-model Validation of Integrated MARTE and Component-Based Methodology Component Model for Embedded Real-Time Software

Mohd Z.M. Zaki, M.A. Isa, and Dayang N.A. Jawawi

Software Engineering Department,
Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia 81310 Johor Malaysia
{zulkiiflizaki, mohdadham, dayang}@utm.my

Abstract. A validation process for integrated model-based methodology for component-based embedded real-time software with a profile is presented in this paper. Unified Modeling Language for Modeling and Analysis Real-Time and Embedded System, as a newly developed profile has been introduced to overcome problems in previous profiles. Nevertheless, a sound and systematic methodology is needed in order to tackle complexity problems that arose. The objective of this paper is to validate the integrated profile and a selected component-based methodology component model for satisfying embedded real-time software requirements, thus helping engineers to model their system, enhancing the structure and component modeling. For that, this paper described a component model meta-model validation process using quality matching for the integration process, involving a profile and a methodology. Nevertheless, this paper focused more towards the validation of the integrated component model before can be implemented on Embedded Real-Time software development, whereby the proposed integration component model is applied on a case study to show its enhancements. The integration result will support to solve complexity whereby the profile is used to solve the lack of specific modeling language notation for embedded real-time system and the method can provide a systematical software process.

Keywords: Model-based Methodology; UML; MARTE; Component-Based Software Engineering; Embedded Real-Time Software; Mapping; Integration; Quality Matching.

1 Introduction

Software engineering discipline has covered almost most of application domain, including Embedded Real-Time (ERT) system. ERT software is unavoidably become more and more complicated and sophisticated. This has causes the need of more acceptable approaches rather than simply rely on a simplex traditional concept of procedural programming that no longer sufficient to overcome very large systems, which can carry a million or more lines of code, the programming level.

ERT software design has become an important research area due to the high complexity of the new generations of such systems [1]. ERT software differs from the traditional data processing systems are that they are constrained by non-functional requirements such as dependability and timing constraint [2]. These special features actually will correspond to the increased of complexity as it derives from the amount of functionality that is associated with those systems [1]. Specification, modeling, synthesis, simulation, and verification are required to cope with this increasing complexity, as the need of efforts in all areas of system level design. In additions, designing and modeling ERT systems are becoming more difficult as the requirements to the systems are increasing due to the user's needs and demands, which direct to complexity problems [3]. Therefore, ERT software development must be able to cope with complexity, to adept quickly to changes and capable to support extra-functionality. Therefore, the use of models can provide a very high level of abstraction on the one hand; on the other hand they are very well suited for visualization [4].

This is where Unified Modeling Language (UML) has become a strong candidate for specifying and designing ERT systems, which is mainly because of the abstraction provided by this language [5]. UML is a graphical language for visualizing, specifying, constructing, documenting and executing software systems [6]. UML was adopted by Object Management Group (OMG) as a fusion among several of the best OO methods such as Booch [7], Harel [8], Jacobson [9] and Rumbaugh et.al [10]. The advantage of UML is that the providence of extension mechanisms, which allow customization of the language, enabling the definition profiles for specific domains [1]. But, UML stills developed and challenged by some important problems, such as the ability to cope with ERT unique requirements like real-time constraints, resource restrictions and modeling independent components [11][12].

Therefore, the use of Unified Modeling Language (UML) profiles for ERT modeling to develop ERT system can aids the development process especially for modeling complex ERT systems [13][14]. From this perspective, a UML for Modeling and Analysis Real-Time and Embedded (MARTE) appears to be the suitable approach for modeling ERT systems [15]. Intends to replaces the previous profile [16] [17], MARTE has provides some new key features such as support for non-functional property modeling and adds rich time and resource models to UML. Therefore, a model-based methodology has been study to suite the need of MARTE, making the purpose of MARTE profile become more extensive and useful.

Yet, there still need a proper validation so the integration meta-model can be validated and measured for confirming the validity and correctness of the integrated component model. Through the validation process, the integrated component model can be measured and analyzed before the implementation. In this study, the integrated component model is implemented in software process. In this study, the validation process is reused from a technique called as Meta-model Matching Experiment [4].

The remainder of this paper is structured as follows: The next section briefly describes the motivation of this paper; Section III describes the validation methodology for the integrated component model; Section IV presents the validation result, based on the calculation of the proposed integration, which involved the study and discuss results. Finally, Section V presents a brief summary and conclusion drawn from this paper.

2 Motivation

However, there is no specific methodology or process with MARTE. Therefore, an integration between MARTE profile and a methodology has been done [1]. This is because, beyond UML and its profiles, another focus is on its methodology and software process, whereby UML specification only specifies syntax and semantics of its notation, but does not determine how to apply its elements within a development process [12]. This is where Method for Component-Based Real-Time Object Oriented Development and Testing (MARMOT) is acceptable to fulfill the needs. Acts as a systematical and sound development method for component-based development (CBD) and model-driven, MARMOT has been derived from Kobra [18] to yield elements of developing ERT software.

The main objective of this paper is to validate the integrated MARTE Generic Component Model and MARMOT Component Model for satisfying embedded real-time software requirements. The process of validation is including a component model meta-model validation process using quality matching for the integration process. This validation process has been performed to validate the correctness of the proposed integrated component model meta-model. There are other techniques for validating meta-model integration but the quality matching technique has been chosen. This is because of this techniques can relatively show how the integration is properly being done based on their elements weight. With correct weightage, the integrated meta-model can be assumed have been properly integrated. Therefore, in order to do the validation, this paper has described a proper and gradually manner to facilitate the validation process.

As results, towards this validation, the proposed integrated component model meta-model, which consists of MARTE profile, will act as the enhancement of the current implementation of modeling language that has been adapted in MARMOT methods. As the purposes of MARTE itself will be used to enhance the elements, especially that are related to the development of ERT software. On other hands, with the integration, MARMOT indirectly has provided MARTE a proper and systematic software process and methodology, which one of the current disadvantages of MARTE profile [1].

3 Validation Methodology

Validation of the integrated MARTE and MARMOT component model meta-model is possible to be done. It is because both of them, although not in the same paradigm, still depend on UML as the core modeling language of both profile and method. This modeling language is the parent of MARTE profile, while has been used as the main modeling language in MARMOT. Based on this reason, the process can relied on the meta-models to validate the integration.

To aid in the validation process, the validation technique is using the Meta-Model Matching Experiment to measure the quality of matching results. This work is focused on the validation of integration meta-model with respect to its quality matching and relevancies numbers of elements after the integration. Hence, in order to measure these

features, a matching meta-model measurement technique is being reused which originate in the field of the information retrieval [4] to compare numbers of MARMOT CM meta-model elements, Mm and MARTE GCM elements, Mt . In this technique, *Precision* and *Recall* are the primary measurements, whereby *Precision* has a mutual relationship with *Recall*, in which one thing affects or depends on another. Then, this technique uses primary measurement, *MM-measure*.

These measures are based on the notion of *true positive* (tp), *true negative* (tn), *false positive* (fp) and *false negative* (fn). Definition of tp is a number of elements that overlap between MARMOT CM and MARTE GCM elements ($Mm \cap Mt$) while fp is a number of elements of MARMOT CM and MARTE GCM elements that are not overlapped with each other. This is called as false matches, $fp = Mm \cap Mt$ where $Mt = (|tn| + |fp|)$. In additions, fn defines the number of MARTE GCM elements that not overlapped with MARMOT CM elements. This is called as missed matches, $fn = Mt \cap Ma$ where $Ma = |fn| + |tn|$. In this situation, true negative, tn represent the elements, which not in MARMOT CM or MARTE GCM.

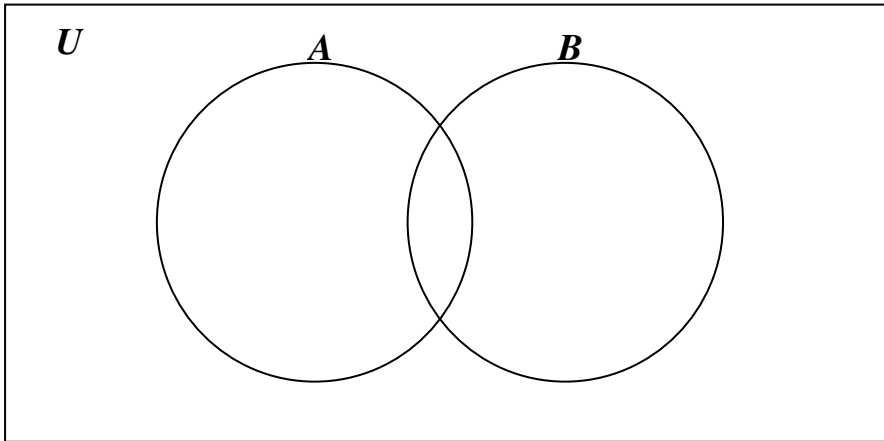


Fig. 1. The Venn diagram related to MARMOT CM and MARTE GCM meta-model integration

Figure 1 illustrates Venn diagram including all measures and interceptions of related sets elements of MARMOT CM and MARTE GCM. Based on the cardinalities of these sets, measurements as mention before are defined by [4]. But, to avoid the faults and miscalculated, the meaning of each terms which has been used by [4], are redefined using the specification from [20] to assured that each terms used in the measurement are satisfied the related elements in the integrated meta-model. This is because the originality of the technique is being used in information retrieval domain; therefore there is a need to redefine to ensure the applicability of the measurement to be used in other domain as well.

$$Precision = \frac{|tp|}{|Ma|} = \frac{|tp|}{|tp| + |fp|}$$

Precision is used to evaluate the relevant matching of the MARTE GCM elements into MARMOT CM elements. If the *precision* is higher, then the matches are found. On other hand, if the number of *fp* is equaled to zero, then all matches are considered to be correct.

$$Recall = \frac{|tp|}{|Mt|} = \frac{|tp|}{|tp| + |fn|}$$

Recall is used to evaluate the frequency of relevant matches compare to the set of relevant matches *Mm*. If the *recall* is higher, then it states that all relevant matches have nearly been found.

$$MM - measure = 2 * \frac{|tp|}{(|fn| + |tp|) + (|tp| + |fp|)} = 2 * \frac{(Precision * Recall)}{(Precision + Recall)}$$

MM-measure uses both *Precision* and *Recall* to encounter any misestimating measurements. Therefore, there still need an equal weight average of *Precision* and *Recall* for formal measurement.

Therefore, in order to evaluate the relevancies number of the integrated component model meta-model, a basic mathematical logic based on set theory has been used based on the rules and guidance in [19]. The mathematical logic equations are modeled as below:

$$Ma \cap Mt = \{X : X \in Ma \text{ and } X \in Mt\}$$

$$Ma \cup Mt = \{X : X \in Ma \text{ or } X \in Mt\}$$

$$Ma \cup Mt = IMM, \text{ where } IMM = Ma \cup Mt = |fp| + |fn| + |tp|$$

Based on the equations, the number of integrated meta-model elements (*IMM*) will be equivalent with the number of *Ma* union *Mm*, $Ma \cap Mt$ and the summation of *tp*, *fp* and *fn*. So, the integrated meta-model are considered as relevant matches if *IMM* have the equalities with $Ma \cap Mt$ and the summation of *tp*, *fp* and *fn*.

4 Validation Results

The integration of two component model meta-models is represented in Figure 2. Using MARMOT component structure as the integration point does integration on both of the component model. This is because as to provide methodology to MARTE, the integration will mostly based on MARMOT method, and then will be enhanced using new elements from MARTE that has been introduced to support ERT systems development, mainly in designing and modeling phase.

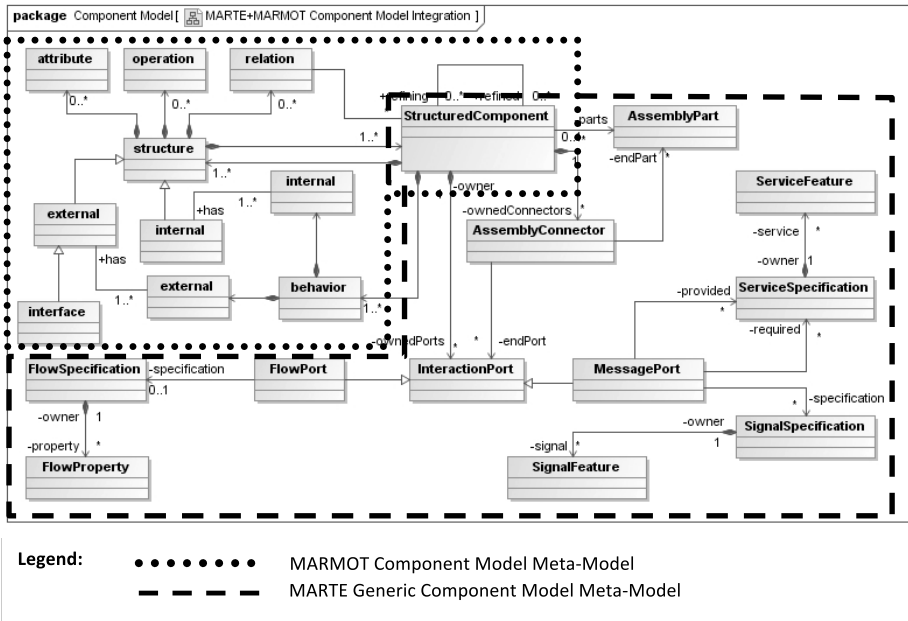


Fig. 2. The Integrated Component Model Meta-Models

During the integration process, there are no elements of meta-models; either from MARTE GCM or MARMOT CM has been removed from the original meta-models structures. Instead, the integration of the meta-models has considered more towards the similarity of the elements, which has been recognizing based on the mapping process. Therefore, the elements in both MARTE GCM and MARMOT CM will remain the same relationships.

Based on the integration, most of the enhancements are involving the capabilities of expressing the constraint requirements of ERT software directly into the modeling. This includes the introduction of new port mechanisms in structure and temporal properties mechanisms in behavior. The application of the integrated component model meta-model will be used at development phases. Based on the integration meta-model, the current component model that has been adapted in MARMOT method will be enhanced. The enhancement will be supported by new elements that have been proposed in MARTE in supporting design and modeling for ERT systems.

The integration of MARMOT CM meta-model and MARTE GCM meta-model consists of 8 elements of MARMOT CM (Ma) and 13 element of MARTE GCM (Mt).

$$Ma = \{ Component, Structure, ExternalStructure, InternalStructure, Behavior, InternalBehavior, ExternalBehavior, Interface \}$$

$$\therefore |Ma| = 8$$

$$Mt = \{StructuredComponent, AssemblyConnector, InteractionPort, \\ AssemblyPart, FlowPort, FlowSpecification, FlowProperty, \\ MessagePort, SignalSpecification, SignalFeature, \\ ServiceSpecification, ServiceFeature\}$$

$$\therefore |Mt| = 13$$

Based on the result of the integration, it is found that 2 elements are overlapped and have been integrated with each other respectively. These elements have been recognizes as *MM*:

$$MM = \{StructuredComponent\}$$

$$\therefore |MM| = 1$$

The value of true positive (*tp*) is equals with the number of *MM* elements, where:

$$tp = |MM|$$

$$\therefore tp = 1$$

The value of false positive (*fp*) is equals with the subtraction of MARMOT CM meta-model elements with *tp* elements, where:

$$fp = |Ma| - |tp| \\ = 8 - 1 = 7$$

On other hand, the subtraction of MARTE GCM meta-model elements with *tp* elements will produce the value of false negative (*fn*) value such as:

$$fn = |Mt| - |tp| \\ = 13 - 1 = 12$$

By using value from *Ma*, *Mt*, *tp*, *fp* and *fn*, the calculation of *Precision*, *Recall* and *MM-measure* can be done. Therefore, the value of *Precision* and *Recall* can be calculate such as the way:

$$Precision = \frac{|tp|}{|Ma|} = \frac{|tp|}{|tp| + |fp|} \\ = \frac{1}{8} = \frac{1}{1 + 7} \\ = 0.1250$$

$$\begin{aligned}
 Recall &= \frac{|tp|}{|Mt|} = \frac{|tp|}{|tp| + |fn|} \\
 &= \frac{1}{13} = \frac{1}{1+12} \\
 &= 0.1333
 \end{aligned}$$

Accordingly, the calculation of *MM-measure* can be done as follow, by utilizing the *Precision* value, 0.1667 and *Recall* value, 0.1333 into the calculation:

$$\begin{aligned}
 MM - measure &= 2 * \left(\frac{(Precision * Recall)}{(Precision + Recall)} \right) \\
 &= 2 * \left(\frac{(0.1250 * 0.0769)}{(0.1250 + 0.0769)} \right) \\
 &= 2 * \left(\frac{(0.0096)}{(0.2019)} \right) \\
 &= 0.0951 \\
 &\approx 0.1000
 \end{aligned}$$

Consequently, the integrated component model of MARMOT and MARTE comprises balance average result, where the *MM-measure* is 0.1000, equally distributed with the average value of *Precision* and *Recall*.

IMM = {*StructuredComponent, Structure, ExternalStructure, InternalStructure, InternalBehavior, ExternalBehavior, Interface, AssemblyConnector, InteractionPort, AssemblyPart, FlowPort, FlowSpecification, FlowProperty, MessagePort, SignalSpecification, SignalFeature, ServiceSpecification, ServiceFeature, Behavior*}

$$\therefore |IMM| = 21$$

The new integrated component model meta-model elements, *IMM* has 21 elements and the number of *Ma* union *Mt* also has 25 elements. In addition, the summation of *tp*, *fp* and *fn* also equals to 25 elements. The detail calculations are shown below:

$$Ma \cap Mt = \{X : X \in Ma \text{ and } X \in Mt\}$$

$$Ma \cap Mt = |MM| = 1$$

$$\therefore Ma \cap Mt = 1$$

$$Ma \cup Mt = \{X : X \in Ma \text{ or } X \in Mt\}$$

$$\begin{aligned} Ma \cup Mt &= |Ma| + |Mt| - |MM| \\ &= 13 + 15 - 2 \\ &= 25 \end{aligned}$$

$$\therefore Ma \cup Mt = 25$$

$$Ma \cup Mt = IMM, \text{ where } IMM = Ma \cup Mt = |fp| + |fn| + |tp|$$

$$\begin{aligned} Ma \cup Mt &= IMM \\ &= |fp| + |fn| + |tp| \\ &= 10 + 13 + 2 \\ &= 25 \end{aligned}$$

$$\therefore IMM = 25$$

$$IMM = Ma \cup Mt = |fp| + |fn| + |tp|$$

if

$$IMM = 25,$$

$$Ma \cup Mt = 25,$$

$$|fp| + |fn| + |tp| = 10 + 13 + 2 = 25$$

then

$$IMM = Ma \cup Mt = |fp| + |fn| + |tp|$$

$$25 = 25 = 25$$

$\therefore IMM$ is relevant match

Therefore, based on the assumptions using relevancies of theory set, the integrated component model meta-model can be assumed as a relevant match.

5 Conclusions

The main objective of using integrated approach in this research is to support MARTE profile with a sound and systematical CBD software process method. In order to integrate these profile and method, two component models are used: MARTE GCM and MARMOT CM. MARMOT has been used as its extensions to support ERT system development using component technology.

This paper showed the flow of validation process of integrated component model of ERT profile and CBD method for ERT by using the proposed methodology. The result of paper is an integrated component model meta-model which map MARTE GCM to MARMOT CM. Besides, the mapping also in directly have detailed up MARMOT CM by adding multi-constraint requirement elements through property bundles from MARTE GCM. The new integration meta-model can support ERT systems with CBSE by considering issues of complexity of ERT software, multi-constraints extra-functionality and systematically software methodology and process for MARTE profile.

In addition, the integration also give the new enhancement during modeling a component, which give more specific modeling artifacts and more details design models, compared if model using UML 2.0. This enhancement can give advantages to the ERT software developers in designing and modeling ERT systems with more specific artifacts and notations.

Acknowledgement

Special thanks to Shahliza Abd Halim, Mohd Adham Isa and all Embedded Real-Time and Software Engineering Lab (EReTSEL), www.kpnet.fsksm.utm.my/eretsel for their knowledge supports and motivations, Ministry of High Education (MOHE) Fundamental Research Grant Scheme (FRGS) fund for the financing and funding and Universiti Teknologi Malaysia (UTM) for the facilities and infrastructures.

References

1. Lisane, B.B., Marcio, E.K., Luigi, C.: UML as front end language for embedded system design. Behavioral Modeling for Embedded Systems and Technologies: Applications for Design and Implementation (July 2009)
2. Stankovic, J.A.: Real-time and embedded systems. *ACM Computing Surveys* 28(1), 205–208 (1996)
3. Jochen, M.K., Joachim, S.: Consistent Design of Embedded Real-Time Systems with UML-RT. In: Fourth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing, p. 0031 (2001)
4. Kappel, G., Kargl, H., Kramler, G., Schauerhuber, A., Seidl, M., Strommer, M., Wimmer, M.: Matching Metamodels with Semantic Systems - An Experience Report. In: Workshop Proc. of Datenbanksysteme in Business, Technologie and Web (BTW 2007), Germany (2007)
5. Lavagno, L., Martin, G., Selic, B. (eds.): UML for real: Design of embedded real-time systems. Kluwer Academic, Dordrecht (2003)
6. OMG. Object Management Group, Unified Modeling Language Specification, v1.0. (1997), <http://www.omg.org>
7. Booch, G.: Object-Oriented Design with Applications, 2nd edn. Addison-Wesley, Reading (1994)
8. Harel, D.: Modeling Reactive Systems with Statecharts. Dorset House, New York (1998)
9. Jacobson, I.: Object-Oriented Software Engineering. Addison-Wesley, Reading (1992)

10. Rumbaugh, J., Blaha, J.M., Premerlani, W., Eddy, F., Lorenson, W.: *Object Oriented Modeling and Design*. Prentice Hall, Englewood Cliffs (1991)
11. Lutz, B., Ansgar, R., Andreas, S.: Evaluating UML Extensions for Modeling Real-Time Systems. In: *Seventh IEEE International Workshop on Object-Oriented Real-Time Dependable Systems (WORDS 2002)*, p. 271 (2002)
12. Douglas, B.P.: *Real Time UML: Advances in the UML for Real Time Systems*. Pearson Education Inc., Boston (2004)
13. Crnkovic, I., Larsson, M.: *UML for Real: Building reliable component-based software systems*. Artech House, Boston (2002)
14. OMG. MARTE specification beta 1. (OMG document.ptc/07-08-04), (2007a), <http://www.omg.org>
15. OMG. UML Profile for Schedulability, Performance and Time (OMG document no.ptc/02-03-02), (2002), <http://www.omg.org>
16. Woodside, C.M., Petriu, D.C.: Capabilities of the UML Profile for Schedulability Performance and Time (SPT). In: *Workshop SIVOES-SPT on the usage of the SPT Profile held in conjunction with the 10th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2004)*, Toronto, Canada (May 2004)
17. Choi, Y., Bunse, C.: Towards component-based design and verification of a μ -controller. In: *11th International Symposium on Component-Based Software Engineering*, pp. 196–211 (2008)
18. Daniel, J.V.: *How to Prove It: A Structured Approach*, 2nd edn. Cambridge University Press, Cambridge (2006)
19. Salton, G., McGill, M.J.: *Information Retrieval. Grundlegendes für Informationswissenschaftler*. McGraw-Hill, New York (1987)

Abstract Formal Framework for Method Overriding

Siti Hafizah, Mohd Sapiyan Baba, and Abdullah Gani

Faculty of Computer Science and Information Technologies,
Universiti Malaya, 50630 Kuala Lumpur, Malaysia
{sitihafizah,pian,abdullahgani}@um.edu.my

Abstract. Automated verification on method overriding is important in Object-Oriented Programming Language (OOPL) to reduce human errors during software verification process. Static verification method fails to address the issue effectively. This paper examines the issues in developing semantics for verifying method overriding in OOPL. The purpose is to identify elements or components for verification process. A study conducted on literature reveals two main issues of verifying method overriding: subtyping and class invariant. Both issues are resolvable by integrating the elements of non-reverification, modularity, and programmer intervention into a framework. We propose an abstract formal framework with the integration of the three elements by using abstract interpretation and Lazy Behavioral Subtyping (LBS). The framework shows that the integration of less restriction of LBS and abstract interpretation is possible to achieve automated verification.

Keywords: program analysis, abstract interpretation, software verification, method overriding.

1 Introduction

Software verification is an important process in software development to ensure the software specifications are achieved. The verification process proves or disproves the correctness or optimization of the semantic of the software by using formal specification of formal methods. In Object-Oriented Programming Language (OOPL), program reuses and re-forms through the use of classes. Method overriding is a concept that is used to redefine method's definition where it affects classes and objects' behavior. The ability of changing of behavior makes the program unstable that leads to unexpected program termination. In the context of method overriding, the unstable program is due to class, method, and data.

Since the presentation of Hoare's seminal paper [7] on data abstraction, class invariant concept has become a popular method to verify programs. In the environment of object-oriented, the method which is called Hoare's Logic has been enhanced to be modular by Muller [14], to control class's specification of data hiding and encapsulation. During verification process of OOPL, programmer encounters problem in object mutability and subtyping. Therefore, Liskov [11] has presented a notion called behavioral subtyping to overcome the problem exists on relationship between methods where it affects on how the methods behave. Over the years,

Parkinson [16] has extended the class invariant by introducing Separation Logic that enables to verify pointer in OOP and subtyping. However, those works needs human intervention by annotation during the verification process. Therefore, Logozzo [12] proposes a framework that verifies OOP by using abstract interpretation theory. Yet, the work on using abstract interpretation on OOP modular analysis and behavioral subtyping concentrates on superclass only. The inference rules are able to avoid the reverification problem; however it results in over-approximation on subclass invariants.

Based on abstract interpretation theory, we design an abstract formal framework by integrating method overriding which involves late bound method calls. Abstract interpretation [5] is used compare to axiomatic or denotational semantic so that the program analysis excludes the requirement of annotation as verification tools [14][3][4][16]. Late binding is included in the framework due to flexible code reuse. However, it complicates the reasoning when the method calls at late binding cannot be statically determined. Programmers use behavioral subtyping to overcome the problem even though it restricts the specification of *precondition is contravariance and postcondition is covariant*. Therefore, the framework of method overriding incorporates Lazy Behavioral Subtyping [6] where the technique supports incremental reasoning and in least restriction manner.

This paper is organized as follows: section 2 discusses related works on invariants and subtyping; section 3 proposes the abstract formal framework by using the model of verifying method overriding and the architecture of abstract formal framework; section 4 presents the general definition of the verifiable semantics of method overriding; section 5 discusses the abstract formal framework. Finally section 6 presents the conclusion.

2 Related Works

Invariant is a concept from Mathematics where it is generally described as value of expression that does not change during the program execution. The purpose is to become a property that is true of all expressions of a given code at all time [18]. Approximate polynomial representation is used for the shape of an invariant. Once the shape of the invariant is predicted, a deterministic technique is used to generate the exact form of the invariant [17]. The idea of a class invariant that is first proposed by Hoare has been extended so that the structure of classes and objects will be inclusion. The class invariant is a difficult task in inheritance because the class invariant is meant for single object, but there is aggregate structure in inheritance that involves two or more objects. Re-verification of superclass every time subclass is added into the program can cause code blow-up. Therefore, Parkinson [15] proposes of using a general foundation of verification which is predicates to specify the properties of aggregate structure, after considering the complexity of peer invariants of [8] and history invariants of [9]. The predicates are represented as Separation logic which is extension to Hoare logic. However, the method involves annotation that can lead to human errors. Based on existing invariants generating techniques, Xing et al. [19] present a technique where invariants are generated automatically at each statement to ensure properties are safe and terminated. The technique does not need to use Hoare

logic and the weakest precondition strategy. Nevertheless, the reasoning codes add up as the program codes increase. Banarjee merges non-computer related technique; clonal selection theory, into program verification process to predict program invariant shape [1]. The method successfully can predict the program invariant correctly on while-program. However, the verification process works as non-modular. The verification codes can extend when the verification processes object-oriented program.

Basic *subtyping* principle is substitutability. The substitution happens among objects or classes. Liskov [10] states $S <: T$, where S is a subtype of T , if a value of type S can be provided whenever a value of type T is required. For example, when type integer is a subtype of type double, $\text{integer} <: \text{double}$, then number 10 that declared as integer can received as double as well. By considering contravariance and covariance [2], substitutability has better notion under behavioral subtyping [11] which is *precondition of method is always be contravariance and postcondition of method is always be covariance*. The emergence of separation logic [16] and behavioral subtyping [11] produce specification subsumption that is able to avoid reverification [4]. The combination of separation logic and behavioral subtyping focuses the distinction and relation between specifications to support behavioral subtyping in class invariant. Both superclass and subclass are verified at the same time by considering their behavioral subtyping and method overriding. However, the method needs programmer to annotate the specification onto the program. Therefore, to avoid human annotation, Logozzo [13] has extended the subtype relation by considering arbitrary class properties. The extension uses modular static analysis by using abstract interpretation and observable behavior of program in order to verify polymorphic code. The inclusion of observable behavior increases the execution time of verification process. In addition to that, the observable behavior is for the program that follows the notion of behavioral subtyping only.

The above related works reveal that all of the works lack one or more of three components in verifying method overriding. The three components of method overriding verifications: non-reverification, modular, and automated are identified that determine the efficiency of the verification process. First is non-reverification. Reverification is a verification of more than one process on the same code. It happens in two segments in which method and class are in inheritance. The relationship is represented as subtyping, when a subclass is added into a superclass. The superclass is verified again every time subclass is verified because there is relationship between them (for example, composition or generalization). Therefore, it makes the process longer than without inheritance. Second is modular. Modular is a software design technique on which level of program is separated through method, class, object, component or loop. The technique gives the advantage of more understandable design, clear relationship between modules, and easily program manipulation. Third is programmer intervention. The verification process needs programmer intervention when the process involved annotation and type system. This can lead to human error during the process. However, it is difficult to make the process automated because program reasoning involved a lot of semantic meanings and prediction of the shape of class invariant.

3 Abstract Formal Framework

We propose an abstract formal framework where the word *framework* means the definitions of semantics of method overriding. Before the framework is produced, analysis on related works has been done. Based on the analysis, there are three features that have to be included. The features are shown in model of verifying method overriding which is illustrated in Figure 1. Non-reverification, modularity, and automated are the three features that determine the safety, code-termination, and reliability of any solution related to checking method overriding program. All of the features are related to each other in the object-oriented process. In order to make the verification process automated, inference rules in the process has to be modular and non-reverification. The aim is to reduce complexity that can affect the performance. Based on the model, architecture of abstract formal framework and semantics of method overriding are produced in later sections.

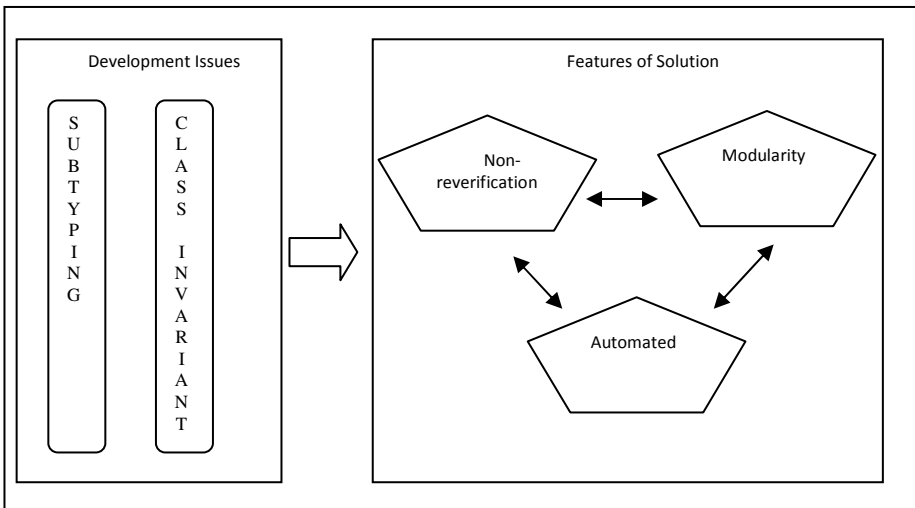


Fig. 1. Model of Verifying Method Overriding

Object-oriented program P can be any language, for example, C++, Java or C#. Every time the program executes, a state Σ of the program is produced. The state means program's execution memory at certain time. The program's state is actually a combination of several local states $(\sigma_i \in \Sigma)^{vi \in 1..n}$ that performs methods (or operations) which can modify state. Overriden methods \mathcal{M} involve more than one method. Methods of subclass enable to modify methods of superclass provided the methods of subclass have same method name and arguments (C++, Java). \mathcal{L}_S enables to characterize the properties of the program P and \mathcal{M} . Based on semantics that captures the program specification, and by considering non-reverification, modularity and automated, the verifiable semantic of the program is defined in $mod(\mathcal{L}_S)$. Therefore, $mod(\mathcal{L}_S)$ is produced by using abstract interpretation theory and lazy

behavioral subtyping. Figure 2 illustrates the architecture of abstract formal framework of method overriding.

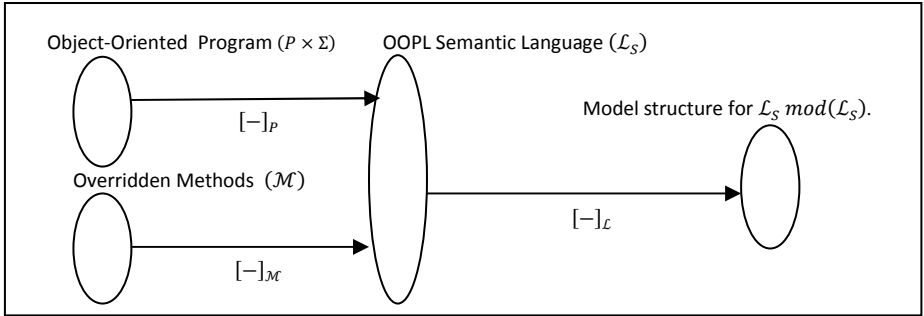


Fig. 2. Architecture of Abstract Formal Framework

Formally, the semantics of the overridden methods \mathcal{M} are given by a function $\llbracket - \rrbracket_{\mathcal{M}}: wff(\mathcal{M}) \rightarrow wff(\mathcal{L}_S)$. It is a map between well-formed formulae of overridden methods $wff(\mathcal{M})$ that are written in certain language and well-formed formulae of semantic language $wff(\mathcal{L}_S)$. In any modern object-oriented language (C++, Java, C#), we assume the overridden methods and the semantic language are written in well-form. The semantic language has the components of specification; non-reverification, modular and automated. Those affect the model structure of \mathcal{L}_S that is explained later in the abstract formal framework. The function $\llbracket - \rrbracket_{\mathcal{L}}: wff(\mathcal{L}_S) \rightarrow \text{mod}(\mathcal{L}_S)$ maps well-formed formulae semantic language with components of specification.

4 Semantics of Method Overriding, $\text{mod}(\mathcal{L}_S)$

This section explains the abstract framework on the semantics part that infers the verification process of method overriding. The inference rules use abstract interpretation to ensure the verification process is automated and Lazy Behavioral Subtyping to ensure there is no reverification on superclass. The verification decomposes onto classes and methods for modularity.

Definition 1 (Class). Class C consists of tuple $\langle \text{const}, f, m \rangle$, where const is the class constructor, f is field declarations, and m is methods. Program P produces state; Σ , which is $\langle E, S \rangle$, where E is environment and S is store. State; Σ consists of many internal states; σ , that come from objects in the program ($\sigma \in \Sigma$). An environment; E , is a map from variables; Var , to memory addresses; A . A store; S , is a map of from addresses; A , to values; Val , where values can be integer, Boolean, null, and reference.

$$\begin{aligned}
 E &\stackrel{\text{def}}{=} \text{Var} \rightarrow A \\
 S &\stackrel{\text{def}}{=} A \rightarrow \text{Val} \\
 \text{Val} &= \{\text{int}, \text{bool}, \text{null}, \text{reference}\}
 \end{aligned}$$

Fig. 3. Concrete Semantics

Definition 2 (Abstract Semantics Domain). Domain of a program is Cartesian product of environment and store; $E \times S$. It has two types which are domain input and domain output; $\langle D_{in}, D_{out} \rangle$. Constructor is a domain input for any class. Its activation produces a state; $\text{const} \llbracket \text{const} \rrbracket \in [D_{in} \rightarrow \rho(\Sigma)]$. A method's abstraction is $m \llbracket M \rrbracket \in [D_{in} \times \Sigma \rightarrow \rho(D_{out} \times \Sigma)]$. In this context, the precondition of the method is a type of object C , where it produces postcondition of type of object collection, B_i . With general form of class C is $\langle \sigma_{\text{const}}, \{m: C \rightarrow B_i\}, \sigma_i \rangle^{\forall i \in 1..n}$, therefore abstract class invariant is

$$C \llbracket C \rrbracket = \text{com} \llbracket \text{const} \rrbracket \sqcap M \llbracket m_i \rrbracket^{\forall i \in 1..n}$$

where $\text{com} \llbracket \text{const} \rrbracket \in \bar{D}$. Abstract domain \bar{D} , is $\langle \bar{D}_{in}, \bar{D}_{out} \rangle$ and $M \llbracket m \rrbracket$ as defined in Definition 6.

Definition 3 (Method of Object). An object is formed from a collection of i methods whose self parameters have type $[m: B_i^{\forall i \in 1..n}]$ which refers to the object itself (distinct). When a method is called through an object $o.m$, the object will call method of its own with precondition x_i with body b_i . In the case of method overriding, the same name of method is updated with different body definition.

$$o \stackrel{\text{def}}{=} m_i = \zeta(x_i) b_i \quad (m_i \text{ distinct})$$

$$o.m \stackrel{\text{def}}{=} \zeta(x_i) b_i$$

$$o.m_i \Leftarrow \zeta(y) b \stackrel{\text{def}}{=} o.m_i \equiv \zeta(y) b = \zeta(y) b_i \Leftarrow \zeta(y) b \quad \text{method overriding}$$

Definition 4 (Typing Rules for Object). E is a static typing environment which is a list of variables with types or variables that they are sub-type of, $E \vdash \diamond$. The type judgment of $E \vdash B_i$ means a collection of B type is a well-formed type in the environment E . The value judgment of $E \vdash b: B$ means b has B type in the environment E .

value x, y

$$\frac{E, x: C, y: C \vdash \text{null}}{E \vdash x: C, y: C}$$

type object

$$\frac{E \vdash B_i \quad \forall i \in 1..n}{E \vdash [m_i: B_i]^{i \in 1..n}}$$

value object

$$\frac{E, x_i: C \vdash b_i: B_i \quad \forall i \in 1..n}{E \vdash [m_i = \zeta(x_i: C)b_i]: C} \quad \text{with } A \equiv [m_i: B_i]^{i \in 1..n}$$

Definition 5 (Typing Rules for Method Object). Let $m_i: B_i$ is a collection of i methods. It produces a type that is behavior to object a . Let object a type of C and object x is type of C can yield b type of B . With both requirements and value for method invocation, method overriding can happen.

value method invocation

$$\frac{E \vdash a: [m_i: B_i]}{E \vdash a.m_i: B_i}$$

Value method overriding

$$\frac{E \vdash a: C \quad E, x: C \vdash b: B_i}{E \vdash \zeta(x: C)b: C \Rightarrow a.m_i}$$

Definition 6 (Semantics of Method, $\mathbb{M}[-]$): Let $D_{in}, D_{out} \subseteq \mathcal{P}(\Sigma)$ and $S \in \mathcal{P}(\Sigma)$. When a method is invoked, abstractly the input of the method is precondition with product of D_{in} that can be nominal variable or object with its state. It produces output D_{out} with state. Method constructor is $\llbracket const \rrbracket \in \mathcal{P}(\Sigma)$. Therefore,

$$\mathbb{M}[-] \in [\mathcal{P}(D_{in} \times S) \rightarrow \mathcal{P}(D_{out} \times S)]$$

The state abstraction of method with input value and output value.

$$\mathbb{M}[\llbracket m \rrbracket](S) = \{\sigma' \in \Sigma \mid \exists \sigma \in S. \exists Val \in D_{in}. \exists Val' \in D_{out}. \langle Val', \sigma' \rangle \in \mathbb{M}[\llbracket m \rrbracket](Val, \sigma)\}$$

Definition 7 (Semantics of Method Overriding). Lazy behavioral subtyping (LBS) uses method use and method declarations to ensure that old proofs remain valid in proof environment. However, in abstract interpretation there is no specification involved. Therefore, the LBS idea is used once overridden method exists in the program. The proof environment is taken from $\mathbb{C}[\llbracket C \rrbracket]$ as \mathbb{I} . Overridden method checks its precondition and postcondition as; $\langle pre_{super} \Rightarrow pre_{sub}, post_{sub} \Rightarrow post_{super} \rangle$. Let $\mathbb{M}[\llbracket m_{new} \rrbracket](S)$ is state of overridden method, therefore general abstract method overriding is

$$\mathbb{I} \sqsupseteq \mathbb{M}[\llbracket m_{new} \rrbracket](S) \quad (1)$$

5 Discussion

The semantics of method overriding in Definition 7 shows the equation is formed by using abstract interpretation theory and LBS. Both theory and technique are used to fulfill the features solution of verifying method overriding. The main focus in the solution is automated. The automated process is crucial in determining class invariant. Therefore, abstract interpretation is chosen to automate the verification process because it is proven can generate inference of class invariants in Java language [12] where the user does not have to annotate the source code by using Java Modeling Language (JML). Beside automation, non-reverification and modularity are also part of verification process. The modularity in the solution means the invariant is in class and method basis. Its purpose is to manage arbitrary aggregate structure of classes. The non-reverification is handled by using LBS. LBS is chosen instead of popular behavioral subtyping because LBS has less restriction rule in the inference rule compare to behavioral subtyping. The method is soundness can be applied as in [20] where the inference system enables to analyze and manipulate the proof environment of multiple inheritance. This is the first attempt (as we know of) of emerging abstract interpretation and LBS to verify method overriding.

The framework is a basis to static analysis of OOPL semantic analyzer. The static analysis on program correctness is done during compilation which is before actual execution at run time. Therefore, the result of the static analysis will be used by the programmer to modify the program code to avoid unexpected failure in the future which syntactic analyzer cannot detect. The semantic analyzer takes place in compiler process where after the source program is analyzed by lexical analyzer and syntactic analyzer. Both lexical analysis and syntactic analysis check the grammar of the programming language using parser. In the situation of object-oriented programming language, the analyzers tokenize the program into classes. Then, the analyzers check the relationship between classes if any and tokenize the language based on its grammar. All information is stored in tables. The analysis of the program proceeds by computing (1) where the computation approximates the semantic of the program based on constructor and methods with data fields. The implementation of (1) which generated from Definition 1 until 6, will be as a library which runs together with other libraries checking the program during compilation.

6 Conclusion

This paper presents a framework of abstract formal for method overriding with the integration of abstract interpretation and Lazy Behavioral Subtyping (LBS) as a result of issues discussed in related work section. The integration enables the class invariant and subtyping being done in modular, without reverification of superclass and most importantly with less human intervention. The framework is initialized from the model of verifying method overriding and the architecture of abstract formal framework. The initialization shows that modularity of class invariant is needed to ensure the inference rules are decomposed into class and method. The decomposition is required to make the reasoning less complex which can affect the performance during the verification. The adaptation of LBS into the framework is to cater the problem of aggregation among classes. In order to consider subclass in the inference system, LBS is helpful especially with its less restriction rules.

References

1. Banerjee, S.: An Immune System Inspired Approach to Automated Program Verification (2009), (February 2011); ArXiv:0905.2649, ArXiv database
2. Castagna, G.: Covariance And Contravariance: Conflict Without A Cause. *ACM Transactions on Programming Languages and Systems* 17(3), 431–447 (1995)
3. Cherem, S., Rugina, R.: A Verifier For Region-Annotated Java Bytecodes. *Electronic Notes on Theoretical Computer Science* 141(1), 183–201 (2005)
4. Chin, W.N., David, C., et al.: Enhancing Modular OO Verification With Separation Logic. *ACM SIGPLAN Notices* 43(1), 87–99 (2008)
5. Cousot, P.: Program Analysis: The Abstract Interpretation Perspective. *ACM SIGPLAN Notices* 32(1), 76 (1997)
6. Dovland, J., Johnsen, E., et al.: Lazy Behavioral Subtyping. *Journal of Logic and Algebraic Programming* 79(7), 578–607 (2010)
7. Hoare, C.A.R.: Proof Of Correctness Of Data Representations. *Acta Informatica* 1(4), 271–281 (1972)
8. M. Leino, K.R., Müller, P.: Object invariants in dynamic contexts. In: Vetta, A. (ed.) *ECOOP 2004*. LNCS, vol. 3086, pp. 491–515. Springer, Heidelberg (2004)
9. M. Leino, K.R., Schulte, W.: Using History Invariants to Verify Observers. In: De Nicola, R. (ed.) *ESOP 2007*. LNCS, vol. 4421, pp. 80–94. Springer, Heidelberg (2007)
10. Liskov, B.: Data Abstraction And Hierarchy. *ACM SIGPLAN Notices* 23(5), 17–34 (1988)
11. Liskov, B.H., Wing, J.M.: A Behavioral Notion Of Subtyping. *ACM Transaction on Programming languages and Systems* 16(6), 1811–1841 (1994)
12. Logozzo, F.: Automatic Inference of Class Invariants. In: Steffen, B., Levi, G. (eds.) *VMCAI 2004*. LNCS, vol. 2937, pp. 211–222. Springer, Heidelberg (2004)
13. Logozzo, F.: An Approach To Behavioral Subtyping Based On Static Analysis. *Electronic Notes in Theoretical Computer Science* 116, 157–170 (2005)
14. Muller, P.: Modular Specification And Verification Of Object-Oriented Programs. LNCS, vol. 2262. Springer-Verlag. PhD Thesis (2002)
15. Parkinson, M.: Class Invariants: The End Of The Road? In: Proceedings of the International Workshop on Aliasing, Confinement and Ownership in Object-oriented Programming, Berlin, Germany, pp. 9–10 (2007)
16. Parkinson, M.J., Bierman, G.M.: Separation Logic, Abstraction And Inheritance. *ACM SIGPLAN Notices* 43(1), 75–86 (2008)
17. Rodriguez-Carbonell, E., Kapur, D.: Automatic Generation Of Polynomial Invariants Of Bounded Degree Using Abstract Interpretation. *Science of Computer Programming* 64(1), 54–75 (2007)
18. Webber, A.B.: What Is A Class Invariant? In: Proceedings of the 2001 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering., pp. 86–89. Snowbird (2001)
19. Xing, J., Li, M., et al.: Automated Program Verification Using Generation Of Invariants. In: 10th International Conference on Quality Software, pp. 300–305. IEEE, Zhangjiajie, ChinaZhangjiajie, China (2010)
20. Dovland, J., Johnsen, E.B., Owe, O., Steffen, M.: Incremental reasoning for multiple inheritance. In: Leuschel, M., Wehrheim, H. (eds.) *IFM 2009*. LNCS, vol. 5423, pp. 215–230. Springer, Heidelberg (2009)

Parametric Software Metric

Won Shin¹, Tae-Wan Kim^{2,*}, Doo-Hyun Kim³, and Chun-Hyon Chang¹

¹ Department of Computer Engineering, University of Konkuk, Seoul, Korea
{wonjjang, chchang}@konkuk.ac.kr

² Department of Electrical Engineering, University of Myonji, Gyeonggi-do, Korea
twkim@mju.ac.kr

³ Department of Internet & Multimedia Engineering, University of Konkuk, Seoul, Korea
doohyun@konkuk.ac.kr

Abstract. Software metrics make meaningful information such as reliability, maintainability to help developers, architecture and so on. It also decides what kind of information is needed to produce a meaningful result from a preset goal. Software metrics should be customizable according to the needs of different users. Earlier, software metrics were not flexible as they supported results focusing on one particular goal. To resolve these problems, we suggest new software metric named Parametric Software Metric. It can be customized by users and it also supports a formula to make a result. Users will get an appropriate result by customizing and using it.

Keywords: Software Metric, Parametric Analysis, Static Analysis.

1 Introduction

Software metrics are needed to understand structure of our software during software development or when we determine costs of maintenance. The software metric has its own software attributes which are used to generate a meaningful result. There have been some researches of making software metric tools for using it easily. However, they have a problem related to reliability. For example, a user uses two metric tools to make decision according to results of tools. However, the user can't decide it because two metric tools have different scopes of software attributes. As a result, they generate different results respectively. On the other hand, a definition of software attribute is ambiguous on each tool [1], [2], [3]. Moreover, it doesn't allow user to define those attributes. To resolve these problems, we define a scope of software attribute clearly and then we generate a formula instead of scalar quantities as a result using those attributes. Users can choose software attributes according to their goal and they can also verify the formula. It means not only users can customize software metric for their goal but also they get an appropriate result from the metric.

2 Related Works

2.1 Software Metric

Software metric is a way to get a meaningful value by using software attributes such as Line of code (LOC), the number of defects and so on. Software metrics can be used

* Corresponding author.

by users when they want to verify whether their software has a good quality or when it is being used to fulfill software requirements, and to know where the improvement is needed in their software [4], [5], [6], [7].

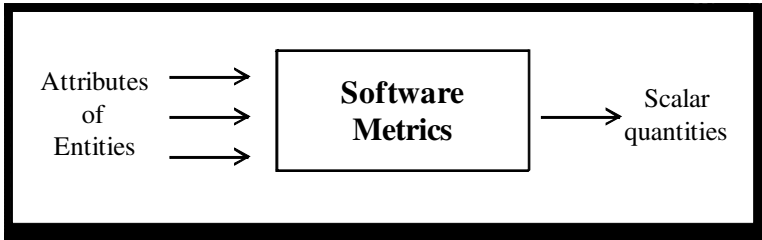


Fig. 1. The concept of Software Metric

In most of the researches, a scope of software attribute has been decided by their criteria. Then, they make a metric based on their software attributes because there is still neither standard of software measures or tools for metrics extraction. A scope of software attribute means a description or a definition of software attribute in detail. For example, LOC may have not less than two meanings. One is the number of all lines of code and another is the number of lines except comment lines.

Determination of the scope of software attribute by their criteria causes two kinds of problems. One is to generate different results because each result is made by calculation of software attributes. It means that the users can make a wrong decision which leads to the increase in the total cost involved as the users use the results of metric as a criteria for making a decision when they resolve some problem in their software [8], [9], [10], [11]. Another one is that users do not trust the result of metric. Users have to know how to make the result of metric and how to decide the scope of software attribute. However, software metric tools do not show the process of making a result and do not give any description of their criteria to decide software attributes. Therefore, it is not available to use software metric for users although they realize its necessity.

To resolve these problems, users choose software attributes so that they can control the software metric according to their criteria.

2.2 Parameter-Based Analysis

Parametric WCET Analysis is one of the Worst Case Execution Time (WCET) analysis methodologies. Previous WCET research has tried to obtain closest value of WCET even without user input so that the results were underestimated or overestimated. To resolve this problem, Parametric WECT analysis defines parameter which is not able to get a value by analysis or estimation from the codes such as the environment variables, or the number of loop bounds. And then it made a formula using the parameter. To get a result from formula, the user needs to input a value of the parameter using an annotation language, dynamic input and a file which contains a value of the parameter. The parametric WCET analysis can support a result by using

the formula which users input as a value of parameter. As a result, users get an appropriate result and it can be applied in diverse environments. Moreover, users can compare and analyze the result by changing a value of parameter [12], [13], [14], [15].

Applying parametric analysis to software metric will help us to resolve the ambiguity of the scope of software attribute and the flexibility of process for making a result in software metric. Moreover, it also helps us to understand how to make a result of metric and what the scope of software attributes are.

3 Parametric Software Metric

A problem of previous researches on software metric is that it is difficult to apply on programs which have different goals. To overcome this problem, a user can decide his own software attributes, which he can use those to make the results. On the other hand, software metric should have flexibility and reliability.

We suggest a new methodology of software metric which named Parametric Software Metric. First we categorize software attributes clearly according to its meaning. Next, we define the segmented attribute as a parameter. Finally, we make a formula by using the parameters. Users input a value of parameters as their goal and they get a good result.

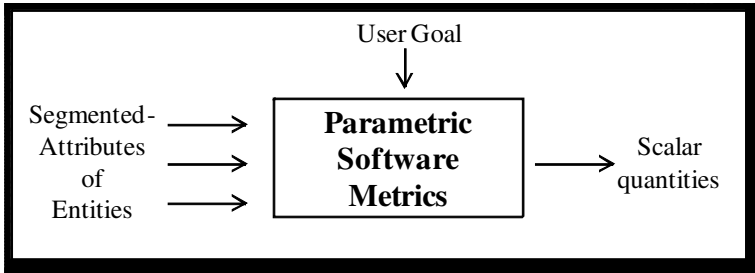


Fig. 2. The concept of Parametric Software Metric

A categorized attributes called Parameter and the formula are characteristics in the Parametric Software Metric. For explanation of Parametric Software Metric, we will show a demonstration of the parametric software metrics by Defect Density Metric (DDM) which is one of the well-known kinds of metrics. DDM is also defined as Quality of software in some researches.

3.1 Organize Items of Parametric Software Metric

The number of defects and LOC are needed to evaluate DDM. DDM is able to find from dividing the number of defects by LOC. In some researches, those attributes are used by different meanings because of its ambiguity. Therefore, we define a scope of those in detail.

In the software life-cycle, some defects are found during the period of testing or reviewing code, and some defects are discovered after releasing software. Therefore, we define two types of defects as below. To use attributes in the formula, we assign the name of attribute such as PREF, POSF and so on.

Table 1. Segmented Attribute of Defect

Attribute	Segmented Attribute	Description
Defect	PRE release Failures(PREF)	A fault discovered during review and testing
	POST release Failures(POSF)	A fault discovered after releasing software

LOC has been used in lots of metrics such as productivity, effort and so on. In previous research, LOC was roughly categorized into comment line and non-comment line. However, we need to separate comment lines from non-comment lines because it means all kinds of codes except comment lines. Non-comment consists of blank line, some lines that contain several separate instructions, and so on. This type of lines has great advantage in readability of code. Some users of metrics tool think that LOC must contain blank line when they want to get the DDM from metrics tool, and vice versa. So we define several types of non-comment line.

Table 2. Segmented Attribute of LOC

Attribute	Segmented Attribute	Description
LOC(Line of Code)	Comment (CLOC)	The number of comment line
	Non-Comment (NCLOC)	The number of all lines of code except comment line
	Blank (BLOC)	The number of blank line
	Line that contains several separate instruction(LSLOC)	The number of line that contains several separate instruction
	Data Declaration (DLOC)	The number of line that contains data declaration

For making a formula, we defined the relationship between segmented attributes. First, CLOC and NCLOC make up the whole lines of code. NCLOC further consists of BLOC, LSLOC and DLOC, they can be included in formula by scope of attribute, and vice versa. There also exist other kinds of segmented attributes and they can be added into scope of attribute, but we do not consider the other attributes in this paper.

3.2 Make a Formula of Parametric Software Metric

We also explain how to make a new formula according to user's goal in this paper. There is a formula for evaluating DDM which is the most commonly used means measuring quality of software. In formula [7], "C" means a part of program code.

$$\text{DDM} = \text{Number of defects discovered in } C / \text{Size of } C \quad (1)$$

The parameter, a size, in the formula can be changed due to the scope of code which is fixed by user. And it also affects a result of DDM. We use LOC for showing example because it has attributes more than the number of defects. It is more suitable as an example. If scope of code means all of code, we can generate a formula as below,

$$\text{Size of C} = \text{CLOC} + \text{NCLOC} \quad (2)$$

If scope of code doesn't contain comments and blank lines, a formula will be as below,

$$\text{Size of C} = \text{NCLOC} - \text{BLOC} \quad (3)$$

If scope of size only contains executable code lines, a formula will be as below,

$$\text{Size of C} = \text{NCLOC} - \text{BLOC} - \text{LSLOC} - \text{DLOC} \quad (4)$$

Developers can get a result they want by substituting the formula (2), (3) and (4) instead of denominator in formula (1). For instance, if scope of size is like formula (4) before software is released, a formula will be as below,

$$\text{DDM} = \text{PREF} / \text{NCLOC} - \text{BLOC} - \text{LSLOC} - \text{DLOC} \quad (5)$$

3.3 Demonstration of Parametric Software Metric

There is the usage through demonstration of Parametric Software Metric using benchmarking code [16] which is usually used for analyzing WCET and also contains all attributes of LOC defined in Table 2. The benchmarking code is bubble sort program named "bsort100".

```

89 BubbleSort(Array)
90 int Array[];
91 /*
92  * Sorts an array of integers of size NUMELEMS in ascending order.
93  */
94 {
95     int Sorted = FALSE;
96     int Temp, LastIndex, Index, i;
97
98     for (i = 1;
99         i <= NUMELEMS-1;          /* apsim_loop 1 0 */
100         i++)
101     {
102         Sorted = TRUE;
103         for (Index = 1;

```

Fig. 3. A part of benchmarking code

Table 3 illustrates the values of each of segmented attributes of LOC and how to compute each kind of lines as a region.

Table 3. Information of benchmarking code

Kind of Line	The number of line (Lines)	A region in Fig 3
Total Lines	128	From 89 to 102
CLOC	23	From 91 to 93
DLOC	21	From 95 to 96
BLOC	23	97
LSLOC	4	From 98 to 100
NCLOC	105	From 89 to 90
		From 94 to 102

To calculate DDM, the number of defects, release or stability boundary is needed. The boundary is a value which is defined by users for comparing with a result of software metric. For instance, it is better to have a low value of DDM. If release boundary is 0.05 and a value of DDM is made by software metric is 0.03, it means that it is possible to release software. Therefore, we suppose that those values are like Table 4 for demonstration.

Table 4. The number of defects and Release & Stability boundary

Item	Value of the item
PREF	5
POSF	2
Release Boundary	0.05
Quality Boundary	0.02

We define two goals first and then we assess quality of software by metrics

(1) **Goal 1** : possibility of releasing software

The purpose of this experiment is finding an answer which is possible in releasing the software. In the experiment, PREF is selected as the number of defects on parameter and the scopes of LOC are used formula (2), (3) and (4).

Table 5. The parameter information of Goal 1

No	PREF	POSF	CLOC	NCLOC	DLOC	BLLOC	LSLOC	Result
1	O		O	O				0.0390625
2	O			O		O		0.0609756
3	O			O	O	O	O	0.0877193

The results of experiments are about 0.03, 0.06 and 0.08 respectively. And the result denotes that it is possible to release software in case of number 1 because it is lower than the release boundary 0.05 as well as it may need many efforts to release in case of number 3 than that of number 2.

(2) **Goal 2** : An assessment of software quality for maintenance

The purpose of this experiment is to expect the costs of maintenance according to DDM. If DDM is too high, it means that it does not assign lots of costs for maintenance. In this experiment, POSF is selected as the number of defects on parameter and the scopes of LOC are used as the experiment of Goal 1.

Table 6. The parameter information of Goal 2

No	PREF	POSF	CLOC	NCLOC	DLOC	BLLOC	LSLOC	Result
1		O	O	O				0.015625
2		O		O		O		0.0243902
3		O		O	O	O	O	0.0350877

The results are about 0.01, 0.02 and 0.03 respectively. It means that the software is stable in test of number 4 because quality boundary is 0.02. Therefore, it may be possible to assign lower costs in case of number 4 than that of number 5 or 6.

Two experiments show the process of arriving at a conclusion in detail. Those also depict that the scope of code can be generated different result from the same code. It means that we can get various results by changing the scope of attributes, and we also find out benefits and harms through comparing different results.

3.4 Characteristic of Parametric Software Metric

There are benefits and harms in our proposed parametric software metric.

(1) Benefits :

Users can customize the metrics according to their goals and then they get appropriate results from the metrics. It means that they trust the result because not only they intervene in the process of metrics but also they know well about that in detail. Moreover, they can know good points or bad points by comparing and analyzing different results by changing the scope of attributes.

(2) Harms :

Users may feel difficult because they need a comprehension of the definition and scope of attribute for using parametric software metric. And they also need to be experienced in using it.

Eventually, the use of software metric we suggested is not friendly and difficult. However, the purpose of using software metric is to get a guideline which leads to good software during software development or maintenance processes. Therefore, it is most important on software metrics to make proper and precise results. The precise results mean that users trust them and know how to do for their software according to them. To get a precise result, software attributes are to be unambiguous and the results of metrics are calculated by using the software attributes which is selected by users. In conclusion, the parametric software metric can resolve those requirements. It means that it is better to use the parametric software metrics than others.

4 Conclusion

In software metrics area, it is so important to support flexibility for users. Though, it was difficult because most of the research in the area of software metrics has concentrated on the precise results. On the other hand, a user couldn't control the software metric, or the user couldn't get appropriate results from it, either. To resolve this problem, we suggest Parametric Software Metric. It helps the users in customizing the software metric according to their goal by supporting some

parameters and formulas. As a result, the users can get a good result by choosing the parameter and input the value of the parameter. And they can avoid making a bad choice through comparing and analyzing the results of metrics.

In future, we will research on categorization of other attributes and develop a tool for improving usability of parametric software metric.

Acknowledgments. This research was supported by R&DB Support Center of Seoul Development Institute, Korea, under Seoul R&BD Program (ST100107) and was also supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency)" (NIPA-2011-C1090-1131-0003).

References

1. Scotto, M., Sillitti, A.: A relational approach to software metrics. In: Proc. of the 2004 ACM symposium on Applied computing, pp. 1536–1540 (2004)
2. Umarji, M., Seaman, C.: Gauging acceptance of software metrics: Comparing perspectives of managers and developers. In: 3rd Int. Symposium on Empirical Software Engineering and Measurement (ESEM 2009), pp. 236–247 (2009)
3. Fenton, N.: New Directions for Software Metrics. In: Keynote Talk CIO Symposium on Software Best Practices, pp. 1–21 (2006)
4. Roche, J.M.: Roche.: Software Metrics and Measurement Principles. ACM SIGSOFT Software Engineering Notes 19(1), 77–85 (1994)
5. Linda, M., Laird, M., Brennan, C.: Software Measurement and Estimations: A Practical Approach. Wiley-IEEE Computer Society Press, Chichester (2006)
6. Fenton, N.E., pfleeger, S.L.: Software Metrics: A Rigorous and Practical Approach, 2nd edn. PWS Publishing Co (1997)
7. Quality Assurance and Metrics, http://www.eecs.qmul.ac.kr/~norman/papers/qa_metrics_artical/index_qa_met.htm
8. Kaur, A., Singh, S.: Empirical Analysis of CK & MOOD Metric Suit. Int. Journal of Innovation, Management and Technology 1(5) (2010)
9. Kaner, C., Bond, W.: Software engineering metrics: What do they measure and how do we know? In: 10th Int. Software Metrics Symposium(METRICS 2004). IEEE Computer Society Press, Los Alamitos (2004)
10. Canfora, G., Concas, G.: 2010 ICSE Workshop on Emerging Trends in Software Metrics. ACM SIGSOFT Software Engineering Notes 35(5), 51–53 (2010)
11. Lincke, R., Lundberg, J.: Comparing software metrics tools. In: Proc. of the 2008 international symposium on Software testing and analysis, pp. 131–142 (2008)
12. Lisper, B.: Fully automatic, parametric worst-case execution time analysis. In: 3rd International Workshop on Worst-Case Execution Time Analysis, pp. 99–102. Polytechnic Institute of Porto, Portugal (2003)
13. Vivancos, E., Healy, C., Mueller, F., Whalley, D.: Parametric Timing Analysis. In: Workshop on Language, Compilers, and Tools for Embedded Systems, vol. 36(8), pp. 88–93. ACM Press, New York (2001)
14. Bernat, G., Burns, A.: An Approach To Symbolic Worst-Case Execution Time Analysis. In: Proc. 25th IFAC Workshop on Real-Time Programming (2000)
15. Coffman, J., Healy, C., Mueller, F., Whalley, D.: Generalizing parametric timing analysis. In: Proc. of the 2007 ACM SIGPLAN/SIGBED Conference on LCTES 2007, vol. 42(7), pp. 152–154 (2007)
16. WCET project, <http://www.mrtc.mdh.se/projects/wcet/benchmarks.html>

Automatic Recognition of Document Structure from PDF Files

Rosmayati Mohemad^{1,2}, Abdul Razak Hamdan¹, Zulaiha Ali Othman¹,
and Noor Maizura Mohamad Noor²

¹ Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600
Bangi, Selangor Darul Ehsan, Malaysia

² Department of Computer Science, Faculty Science and Technology, Universiti Malaysia
Terengganu, 21030 Kuala Terengganu, Terengganu Darul Iman, Malaysia
{rosmayati,maizura}@umt.edu.my, {arh,zao}@ftsm.ukm.my

Abstract. Web-based technology disseminates and stores abundant information electronically. Today, people is more comfortable to use document in Portable Document Format (PDF) because of its operating system independent. However, information on PDF document which in read-only mode are applicable only for human reader. In addition, PDF consists of non-tagged internal structure which make the extraction task difficult. Automatically details analyzing and recognizing of PDF document structures especially paragraph and tabular area is vital for extracting relevant information precisely for use in other domain applications. A combination of heuristic and rule-based approach is proposed to automatically identify and recognize the structure of PDF document. An experimental study has been conducted using a collection of construction tender documents in PDF to test the performance of the proposed approach. The accuracies of precision, recall and f-measures have shown significant results when detecting tabular and paragraph structure.

Keywords: Document Analysis, Information Extraction, Portable Document Format.

1 Introduction

The development of Web-based technology allows abundant information to be kept in digital forms and spread them electronically. Recent study estimated the fluctuation of digital information up to 1610 billion gigabytes in 2011 and about, 94% of this information is unstructured and vague [1]. Organizations are much benefited if they could use 80% of their unstructured resources which stored as text [2].

Various types of digital documents either newspapers, magazines, catalogues, reports, journals and even forms are in Portable Document Format (PDF). Today, people are more comfortable to publish information (text, image, drawing primitives like lines, rectangles) in PDF because it offers open standard feature for sharing, archiving and retrieving electronic document. However, information represented in PDF format is unstructured and inconvenient for reused in another application, for example decision-making application, which requires machine readable information

[3, 4]. Furthermore, there is lack of information on the internal structure of the PDF document content [5, 6]. Thus, details analyzing and recognizing PDF document structure automatically is vital for extracting and decomposing relevant information both into structure and semantic forms for various purposes such as effective sharing, information searching, decision making and others.

Analyzing and extracting particular text with related structure from PDF document is a non-trivial task due to the existing of various different document layout and structure. A block of text element is defined by its underlying document structure tags such as headings, paragraphs or rectangular boxes (tables) [7]. Normal structure of PDF documents generally may consist paragraph, tabular and image. Zanibbi et al. [8] defined table taxonomy as column, row, cell, block, headers, body and associated text regions. Organized texts either in tabular or paragraph ease human understanding and simplify interpretation. However, machine could not identify the structure of tabular form and related text within the table automatically for further processing. Motivation of this study is to enable the machine reading and extracting text similar to human view as much as possible. This paper proposed an intelligent approach to identify and recognize automatically the layout and structure of PDF documents together with their text. The study is carried out in a series of steps to achieve the goal.

The article proceeds in the following manner. Literature on current available tools and past researches regarding on PDF document analysis is briefly reviewed in Section 2. Meanwhile, Section 3 presents on the series of steps proposed to analyze the structure and content of PDF document. Next, the experimental setup is provided in Section 4 and Section 5 discusses on the experimental results analysis. Finally, Section 6 concludes with summary of this research.

2 Related Works

PDF document analysis to recognize the fundamental structure of document is the significant research area that received considerable attention from several researchers. Most of them focused on recognizing tabular structure of document. Therefore, different computational approaches have been applied in this research area such as predefined structure models [9], heuristic approach [6, 10], statistical approach and combination of both heuristic and statistic [11]. In addition, most of this prior researches converting extracted tabular structures and related elements identified on PDF files into other structured format such as HTML and XML format. Hassan [9] proposed wrapper-based approach when detecting table in PDF documents while Jiang and Yang [6] proposed a method to detect PDF document layout and translating it into HTML format. In contrast to our study, recognized text and structure are organized into ontological-based approach which allows for further reasoning process. However, we do not intend to include this part in this paper.

Study on this paper was inspired by the research that had been done by Oro and Rufollo [4, 11] where they had proposed PDF-TREX, a heuristic approach for table recognition and extraction from PDF documents. Here, they carried out analysis on spatial distribution of white spaces between texts for computing horizontal and vertical threshold values to ensure correct text elements were grouped into the same cluster. Threshold value is the fundamental parameter in clustering and the drawback of proposed space distribution analysis is it could cluster dissimilar texts together when the distances

among texts were under predetermined threshold. Therefore, we proposed more details analysis on interpreting appropriate distances between texts by identifying the smallest distance measure, thus minimizing in producing wrong cluster. In addition, we proposed rule-based approach when detecting table. The goal of this paper shares the same interest in identifying tabular structure of PDF documents, yet expands to identify paragraph structure and text associated to tabular and paragraph.

There are currently several commercial products available for analyzing PDF documents such as PDF-Analyzer [12] and PDF Analyzer [13]. Schmoekel [12] developed PDF-Analyzer which offers automatic task for retrieving PDF document internal properties, modifying document security level and basic text or content extraction. The tool however does not capable to recognize any paragraph or tabular structure of the document. Meanwhile, PDF Analyzer released by Amyuni Technologies is most similar to the former product in which capable to identify the PDF document internal properties and has been designed to focus more on the consistency checking of the document structure and content defined by some custom rules. Nevertheless, the tool required complicated and skilled techniques to define custom rules and understand the analyzed result. Complicated usage of this tool makes it more suitable for skilled users (developers) than general end user. In addition, the technology also allows for locating and extracting specific text strings, but there is no specific function that able to understand, infer and extract meaningful information in a table, for instance the relation between text in table header and table body.

3 Proposed Approach

The fundamental challenge in PDF document analysis is to ensure the approach efficiently recognizes and classifies textual elements associated with or within their appropriate structure and layout especially for tabular structure analysis. The orientation of textual elements and possible paragraph and tabular structures are identified through a series of steps as depicted in Fig. 1. Several steps of this approach are inspired by research that has been done by Oro and Rufollo [11].

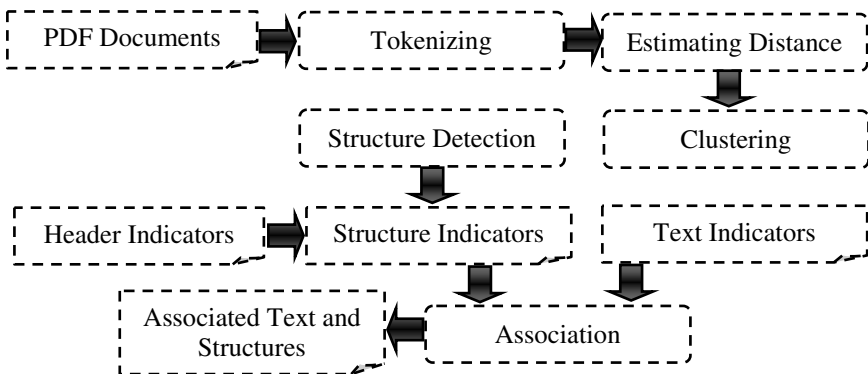


Fig. 1. Series of Steps in Recognizing Structure and Content of PDF Documents

3.1 Tokenizing

Tokenizing is the process of returning text elements or strings identified on PDF documents into tokens together with their absolute 2-dimensional Cartesian coordinates in the original document. A token is defined as non space character. The coordinates for each token, i are identified as upper-left (X_{iL}, Y_{iL}) and bottom-right (X_{iR}, Y_{iR}) coordinates as portrayed in Fig. 2. Here, a token represents a word in the highlighted border. The implementation of this step is accomplished by using Java PDF Extraction Display Access Library (JPEDAL) that reads and identifies all PDF objects. In addition, it also returns the rectangular coordinates of each page of PDF document.

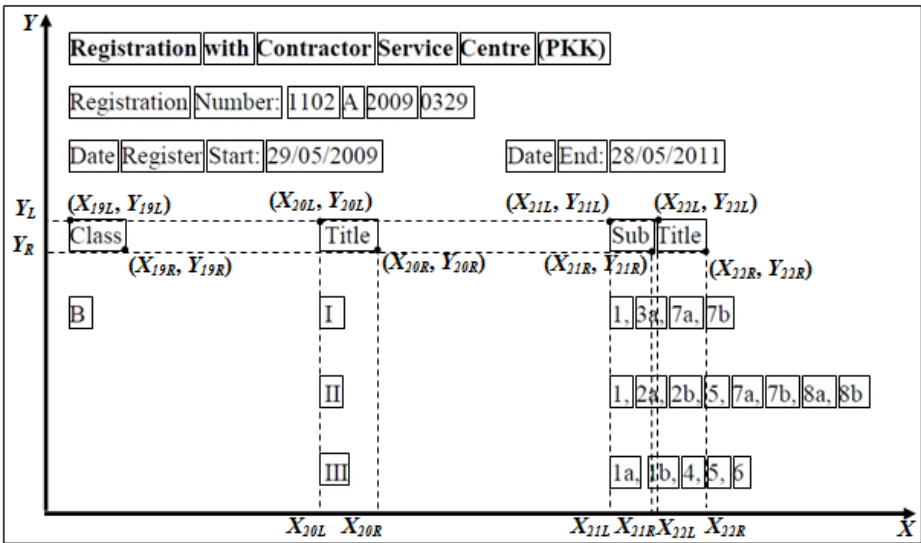


Fig. 2. Text Elements Defined as Tokens with Upper-Left and Bottom-Right Coordinates

3.2 Estimating Distance

Distance estimation is the process to measure space distribution between each token and the closest token to it (nearest neighbour) horizontally and vertically. Space distribution analysis is the most vital stage for computing distance threshold values that will be used in the next step, clustering. Range of distances among tokens are computed separately line by line. This is due to the possibility to have different space distribution in different horizontal lines when the appearance of tokens are affected by different alignment properties, either left, center, right or justify. Thus, horizontal distance between tokens in the same line are calculated as below.

$$DH_{ij} = |X_{iR} - X_{jL}|, \text{ for } i = 0, 1, \dots, n-1 \text{ and } j = i+1 \quad (1)$$

DH_{ij} represents the distance between token i and token j , whilst X_{iR} and X_{iL} denotes the bottom-right X-coordinate of token i and upper-left X-coordinate of token j respectively. The distance estimation is measured by identifying the smallest distance among tokens distributed in the whole page of document. Then, details analysis on the minimum distance is done using statistical analysis. If the difference between minimum distance and other computed distances produces significantly minimum standard deviation error, then the computed distance is defined as the threshold value. Therefore, the number of threshold values determined are based on the number of horizontal lines identified. Meanwhile, the range of distances among vertical lines are also identified in the same way where the distance between vertical line and it neighbour are measured as shown in the equation below.

$$DV_{ab} = |Y_{aR} - Y_{bL}|, \text{ for } a = 0,1,\dots,n-1 \text{ and } b = a+1 \quad (2)$$

DV_{ab} represents the distance between vertical line a and vertical line b , whilst Y_{aR} and Y_{aL} denotes the bottom-right Y-coordinate of vertical line a and upper-left Y-coordinate of token b respectively.

3.3 Clustering

Agglomerative hierarchical clustering algorithm is implemented to group the closest tokens located in the same horizontal line into a cluster. The main purpose of clustering is to group the nearest tokens into a similar block. Initially, the process starts with each token as a separate cluster, c and both upper-left and bottom-right coordinates are defined as the cluster centroid. The number of clusters is reduced by

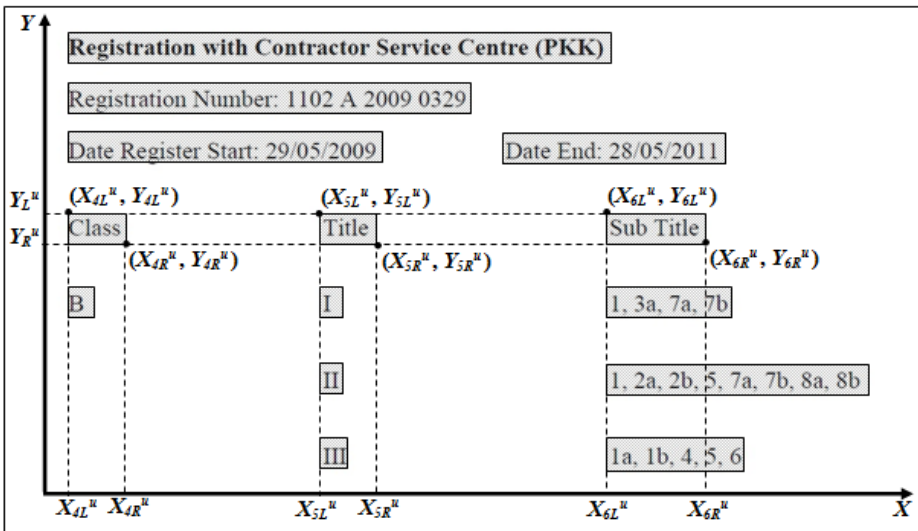


Fig. 3. Clusters Produced with their Centroid Coordinates

merging two clusters with the most minimum distances within horizontal threshold value identified in prior step. Then, the updated upper-left (X_{CL}'' , Y_{CL}'') and bottom-right (X_{CR}'' , Y_{CR}'') centroid coordinates are determined for new cluster created as displayed in Fig. 3. The remaining clusters are continuously merged until there is no more non-clustered element left. The clustering process is repeated for other horizontal lines as well. In other way, similar hierarchical clustering algorithm also applied to cluster the nearest group of tokens vertically. The output from this step is a group of text indicators in which represented by clusters produced.

3.4 Structure Detection

Depending on the number of clusters produced in the previous step, PDF document structure is detected for any existing paragraph and tabular layout. At first, each of the horizontal line is tagged either as non-table or table depending on the structure detection rules described as follow. The line is assumed as non-table line when meet one of the following rules; i) if one cluster or one block is created for the line and length of the cluster is greater than half of document length horizontally, or ii) if one cluster is created for line and length of the cluster is smaller than half of document size and is the first line of the document, or iii) if one cluster is created for the line and length of the cluster is smaller than half of document size and the closest above and below lines exist in the same cluster vertically is defined as non-table line, or iv) if one cluster is created for the line and length of the cluster is smaller than half of document size and is the last line of the document and the closest above line exists in the same cluster vertically is defined as non-table line.

Otherwise, the line is assigned the opposite tag, table line when meet one of the following rules; i) if more than one cluster is created at the particular line, or ii) if one cluster is created for the line and length of the cluster is smaller than half of document size and the closest above and below lines exist in the same cluster vertically is defined as table line, or iii) if one cluster is created for the line and length of the cluster is smaller than half of document size and is the last line of the document and the closest above line exists in the same cluster vertically is defined as table line. After performing these rules, paragraph and tabular structure in the document could be identified. Overlapping clusters horizontally and vertically corresponding to the table line identified is considered to reside in the same row and in the same column for the similar table. Row and column headers are detected by matching with the predetermined header keywords. Meanwhile, non-table line identified is considered as paragraph.

3.5 Association

The final step is to identify and associate appropriate text elements with any recognized paragraph and tabular. Initially, text located within the paragraph is defined belong to the paragraph. Text existed within recognized tabular structure is belong to the table itself. Meanwhile, to significantly describe about the recognized table, in this case title of table, the nearest paragraph to the tabular structure is considered as possible text that describes about the particular table.

4 Experimental Setup and Implementation

The purpose of this experiment is to evaluate the performance of the proposed approach according to precision and recall measurements. A collection of tender documents in PDF for similar building construction project based on Malaysia Construction Tender are used as the experimental data. The total pages of these tender documents are 289 pages. Each page may contain various different layout and structure, for example different size spacing, different type of text alignments (left, center, right, justify). The information on these documents are visually represented in text-based full sentences, form-based and table.

There are two different ways to create PDF document, 1) scanning the existing document using Optical Character Recognition (OCR) and 2) converting documents to PDF using existing tools such as Microsoft Word 2007 onward, PDFWriter, GhostScript and other commercial tools. PDF documents generated from both methods are different in which the former considers the whole content of PDF document as image and the later converts the objects into text and images. In this research, we used PDF documents generated from Microsoft Word 2007. In addition, the experiments are run in Java-based environment and divided into three strategies according to information types.

5 Results and Discussion

Several series of experiments were run to extract the paragraph and tabular structure together with their associated text elements. In order to evaluate the extraction accuracy results, standard information extraction method of precision (PM), recall (RM) and f-measure have been applied. The standard formula for these measurements are shown below.

$$PM = \frac{R}{R + I} \quad (3)$$

$$RM = \frac{R}{R + N} \quad (4)$$

$$f - Measure = \frac{2 * PM * RM}{PM + RM} \quad (5)$$

R is denoted as the number of structures recognized that are relevant, I represents the number of irrelevant structures recognized, and N is the number of relevant structures not recognized. Table 1 shows the comparison results of these evaluation methods for computerized extraction. Two parameters that have been evaluated are tabular structure (row and column detection) and paragraph detection with their associated text. The evaluation of precision, recall and f-measure have shown significantly good accuracy in detecting relevant information. The precision, recall and f-measure percentage rates for tabular structure detection are 76.8 %, 84.0 % and

80.3 % respectively. Meanwhile, the accuracy of paragraph detection have achieved significantly higher results in which 99.4 % of precision, 96.5 % of recall and 97.9 % of f-measure.

Table 1. Comparison Results of Tabular and Paragraph Structure Recognition based on Precision, Recall and f-Measure

Parameters	Computerized Information Extraction Measurements		
	Precision	Recall	f-Measure
Table	76.8 %	84.0 %	80.3 %
Paragraph	99.4 %	96.5 %	97.9 %

The difference of test accuracy between tabular and paragraph is due to the complex structure of tabular format itself compared to paragraph. The finding shows that proposed approach is significantly capable in recognizing the structure of PDF documents, focusing on tabular and paragraph.

6 Conclusion

This study proposed an approach for detecting and recognizing PDF document structure, focusing on paragraph and tabular, then associated text using a combination of heuristic, rule-based and predefined indicators. An experimental study involves a collection of construction tender documents. Based on the evaluation measures of precision, recall and f-measure, the result has shown significant performance when recognizing the document structure and associated text. Current work in this paper is based on simple table assumption. Therefore, the future plan of this research is to include complex table (inner columns and rows) analysis.

References

1. Gantz, J.F., Chute, C., Manfrediz, A., Minton, S., Reinsel, D., Schlichting, W., Toncheva, A.: The Diverse and Exploding Digital Universe: An Updated Forecast of Worldwide Information Growth through 2011, IDC White Paper, vol. 2009 (2008)
2. Froelich, J., Ananyan, S.: Decision Support via Text Mining. In: Burstein, F., Holsapple, C.W. (eds.) Handbook on Decision Support Systems 1., pp. 609–635. Springer, Heidelberg (2008)
3. Rosmayati, M., Abdul Razak, H., Zulaiha, A.O., Noor Maizura, M.N.: Ontological-based for Supporting Multi Criteria Decision-Making. In: Wen, D., Zhou, J. (eds.) 2010 2nd IEEE International Conference on Information Management and Engineering, vol. 1, pp. 214–217. IEEE Press, Chengdu (2010)

4. Oro, E., Ruffolo, M.: XONTO: An Ontology-Based System for Semantic Information Extraction from PDF Documents. In: 20th IEEE International Conference on Tools with Artificial Intelligence 2008, pp. 118–125 (2008)
5. Liu, Y., Bai, K., Mitra, P., Giles, C.L.: Improving the Table Boundary Detection in PDFs by Fixing the Sequence Error of the Sparse Lines. In: 2009 10th International Conference on Document Analysis and Recognition, Barcelona, Spain, pp. 1006–1010 (2009)
6. Jiang, D., Yang, X.: Converting PDF to HTML Approach Based on Text Detection. In: 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, pp. 982–985. ACM Press, New York (2009)
7. Harvey, G.: Adobe Acrobat 6 PDF For Dummies. Wiley Publishing, Inc., Indiana (2003)
8. Zanibbi, R., Blostein, D., Cordy, J.R.: A Survey of Table Recognition: Models, Observations, Transformations, and Inferences. *International Journal on Document Analysis and Recognition* 7, 1–16 (2004)
9. Hassan, T., Baumgartner, R.: Table Recognition and Understanding from PDF Files. In: International Conference on Document Analysis and Recognition (ICDAR 2007), pp. 1143–1147. Curitiba, Brazil (2007)
10. Yildiz, B., Kaiser, K., Miksch, S.: pdf2table: A Method to Extract Table Information from PDF Files. In: Indian International Conference on Artificial Intelligence, India, pp. 1773–1785 (2005)
11. Oro, E., Ruffolo, M.: PDF-TREX: An Approach for Recognizing and Extracting Tables from PDF Documents. In: 10th International Conference on Document Analysis and Recognition 2009, pp. 906–910. IEEE Computer Society, Barcelona (2009)
12. Schmoekel, I.: PDF-Analyzer Pro 4.0., Vol. 1. Software-Development and Distribution, Achim-Uesen, Germany (2010) 1-11
13. Amyuni, T.: PDF Vol. 2010. Amyuni Technologies Inc., Montreal, Canada (2010)

Comparative Evaluation of Semantic Web Service Composition Approaches

Radziah Mohamad and Furkh Zeshan

Department of Software Engineering, Faculty of Computer Science and Information Systems,
Universiti Teknologi Malaysia (UTM), 81310 Skudai, Johor, Malaysia
radziahm@utm.my, farrukh05@hotmail.com

Abstract. As web services are gaining more popularity over the web, there are multiple web services available for different tasks. At run time, the composition of these services based on the requester's functional and non-functional requirements is a difficult task due to the heterogeneous nature of results of the services. This paper introduced some requirements that when fulfilled, a successful composition process can be achieved. In order to find the best approach, various composition approaches on these requirements were evaluated. Suggestions were provided on what approach can be used in which scenario in order to gain the best results.

Keywords: Web Services, Semantic Web Services, Composition Approaches.

1 Introduction

Web services are well defined, self described and reusable software components that can be used over the web using the most silent and stable technologies such as Simple Object Access Protocol (SOAP) as a communication framework, Web Service Definition Language (WSDL) and Universal Description, Discovery and Integration (UDDI) that provides a mechanism to clients to find services [4, 32]. A web service is a set of related functions that can be accessed through programming over the web [14]. The key feature of the web services is that they are loosely coupled, allows ad hoc and dynamic binding and are reusable software components. Web services can be divided into three categories and three entities. The categories are publish, find and bind, while the entities are service requester, service provider and the registry. The roll of facilitator of service outsourcing is one of the most significant aspect of the web that can reduce the overhead of companies and flourish the business [17, 33]. WSDL is the emerging language for describing the present web service technology [17] and presents the syntactic description of the web services. It only present the structure of the data sent and received through the web, but is unable to present the meaning of the data. This makes the automated web service composition difficult as composition, semantic description and execution of web services is necessary for automatic discovery. Existing techniques for web services provides only the syntactic description which as a result, makes it difficult for requester and provider to interpret the meaning of the input and output. Semantic web services are the combination of

web services and the semantic web. In the domain of semantic web, Web Ontology Language for Services (OWL-S) [35] and Web Service Modelling Ontology (WSMO) [36] are two prominent techniques used for service composition. Semantic web services are the extension of the existing web services where the information is represented in a well-defined way [11]. Large amount of data over the web is understandable only by the humans and the custom software [11, 15, 16]. The target goal of semantic web is the medium where the data could be shared easily and processed automatically. The key technology for such concept are the web services [12, 13]. Semantic web services are used for combining data and services from different sources without losing their meaning. Through the discovery and assembly of web services, semantic web services provide the value-added services to complete the domain tasks. [34]. Web services can be combined to provide the unified service with some additional extra values. Four steps are necessary for successful composition of services [37]: (i) Data and control flow model among entities should be created; (ii) For process activities, the services that discover the matched service with the criteria form the service registry should be bounded; (iii) In order to be available to the clients, the composite services should be published in UDDI; (iv) Control and data flow should be managed during the composite service invocation. Although semantic web is gaining popularity, the supporting technologies are still far from the final product, making it an emerging field of research [3].

Since the last decade, considerable work has been done on semantic web services composition [2, 22, 23, 24, 25, 26, 27, 38] but more research is required to address the issue of heterogeneity in automatic (minimal user intervention) web services composition as web services provided by different companies (having their own business rules) provide heterogeneous results. In this paper, we compared and categorized such approaches into two categories; semantic web composition approaches with Quality of Services (QoS) support and semantic web composition approaches without QoS support. The objective of this paper is to identify the best approach that can be used for semantic web services composition. The rest of the paper is organized as follows. Section 2 presents an overview of web service composition approaches. In Section 3, comparative evaluation criteria are discussed. Finally, we present the comparative evaluation remarks in section 4 and the conclusion in section 5.

2 Semantic Web Service Composition Approaches

Semantic web service composition is a widely-studied field since the last decade and different composition techniques have been identified [5, 6, 7, 8, 9, 10]. Most of the work done on semantic web service composition approaches, can be classified into two categories; Semantic web composition approaches with QoS support and Semantic web composition approaches without QoS support. Short descriptions of these approaches are given below.

2.1 Non-QoS-Based Approaches

Sohrabi et al. Approach [20] proposes the introduction of preferences to planning with Golog (the agent programming language). User preferences and the web service descriptions are expressed using a first-order language and are used in a modified version of a Golog interpreter. Evaluation results illustrate the effectiveness of introducing preferences to find optimal compositions.

Aydin et al. Approach [21] proposes event calculus for solving the web service composition problems. With the help of abduction theorem and event calculus (obtained from domain), proper composition plan which corresponds to the user specific composition of web services can be generated. An abduction theorem generates a series of events as well as a set of temporal ordering predicates, giving partial ordering of events which are more suitable for web service composition.

Rao et al. Approach [19] presents a mixed initiative framework for semantic web service discovery and composition allowing user intervention in key decisions. The composition engine combines Web Ontology Language (OWL) ontologies with Jess with a planning functionality based on the GraphPlan algorithm. GraphPlan provides reachability analysis to determine whether a given state can be reached from another state and disjunctive refinement. Planning is used to propose composition schemas to the user. According to the authors, this is the most realistic approach.

Lecue et al. Approach [18] uses Description Logic (DL) reasoning techniques and integrates them with an extended Golog interpreter that can compute conditional Web service compositions and is able to elaborate a strategy for automated branching by means of causal links and laws.

2.2 QoS-Based Approaches

CLM+ Approach. The author in [28] defines five different types of causal links between the input and output of services: (i) Exact (when the input and output parameter are conceptually equivalent); (ii) PlugIn (output parameters are sub concepts of the input parameters); (iii) Subsume (output is a super concept of the input); (iv) Intersection (if the intersection of the output and input are satisfiable); (v) Disjoint (the input and output are incompatible). Causal Link Matrix (CLM) can be developed from a set of services which only considers the functional properties. The author extends CLM to CLM+ to support the non-functional properties. After receiving a request, the most suitable service is discovered from the service repository, and then the semantic connections between services are stored on CLM+ which can be used to compute the composition that represents the possible service composition that matches the service request.

PAWS Approach. Processes with Adaptive Web Services (PAWS) [29] deploys the annotated Business Process Execution Language (BPEL) process with global and local constraints that refer to the QoS aspects. Service Level Agreement (SLA) is used to express the constraints. Through SLA negotiations, advance service retrieval module finds the best service that has the required interface and does not violate the constraints for each task in the created process. If no service interface matches the requirement, then the mediator resolves the service interface descriptions. Multiple candidate services are selected for each task and for each process; only one candidate

service is executed in the BPEL engine. PAWS also allow the faulty services to be replaced with other candidate services and the recovery actions to undo the results of the faulty services.

Zeng et al. Approach. In [31], a goal-directed service composition and optimization framework is presented. The goal directed problem takes three inputs, the domain specific composition rules, the description of the business objectives and the description of the business assumptions. The initial step is called Backward Chaining where the composition rule creates a chain backwards from the business objectives until the initial state is reached and there are no more rules. The second step is called Forward Chaining, where in this step, some additional services are added to the composition schema produced during the first step to complete it, which may be required by the results of some tasks. The contribution of the previous steps contribute to the control flow aspects of the composition. In the final step, called Dataflow Inference, data flow is added to the composition schema.

Roy and Michael Approach. In [30], the authors utilize the semantic web service languages with the model driven methodology to build composite web services. The methodology guides the developers in four phases. At the end of the first phase, an abstract composite model is obtained, containing all the necessary information that can be used for service discovery and selection. In the second phase, suitable web services are handled where the discovery process depends on the semantic descriptions matchmaking. In third phase, a finalized concrete composite model is obtained (for new composite web service, data transformation step is introduced between the services to handle the mismatch between the output of one service and the input of the other service). In the fourth and last phase, different descriptions of the concrete composition model are used for the composed service.

3 Comparative Evaluation

While reviewing different semantic web services composition surveys and approaches [1, 2, 22, 23, 24, 25, 26, 27, 38], we have identified six different kinds of requirements (dynamic aspect, adaptability, domain independence, correctness, scalability and non-determinism) that have large impact on automatic web service composition. These are the minimum requirements that must be satisfied in order to make the composition process successful. The criteria are used to check how far the existing approaches support web service composition. The introduction to these requirements is as given below.

Dynamic Aspect: Dynamic aspect ensures that after a composition schema is designed, it will remain consistent and will be executable for long time.

Adaptability: Adaptation is the process of modifying service-based applications according to the new requirements of the changed environment as specified in the predefined adaptation strategies.

Domain Independence: The ability of composition approach to be applicable on different domains in order to solve the problems.

Correctness: Correctness is the process of checking to ensure that certain properties of the composition maintained.

Scalability: To attain the maximum scalability in terms of performance, limitations must be addressed.

Non-Determinism: Non-determinism may increase the number of composition schemas for a request and must be considered while designing.

4 Comparative Evaluation Remarks

In this section, we have compared the semantic web composition approaches based on the identified requirements (in section 3). We tried to find out their limitations as well as the best category of approaches that satisfies the criteria as completely as possible. Table 1 summarizes the findings.

Table 1. Web service composition approaches

Parameters \ Approaches	Dynamic Aspect	Adaptability	Domain Independence	Correctness	Scalability	Non-Determinism	Semantic Capability	QoS Awareness
PAWS Approach [29].	√	√	√			√	√	√
Roy and Michael Approach [30].	√						√	√
CLM+ Approach [28].	√		√				√	√
Zeng et al. Approach [31].				√			√	√
Aydin et al. Approach [21].			√				√	
Sohrabi et al. Approach [20]			√		√		√	
Rao et al. Approach [19].				√			√	
Lecue et al. Approach [18].			√				√	

The limitations of these approaches are given below.

- **Scalability:** Even if fulfilled, this requirement does not ensure that a composition approach for a given set of effectively working services will also work for a different set of services. The performance of composition approaches should also be tested and the parts that affect the performance must be tuned to ensure the maximum scalability for the given performance. Sohrabi et al. [20] is the only approach that fulfills this requirement.
- **Non-Determinism:** Refers to a case where for a given request, different composition schemas may be obtained. Non-determinism should be considered while designing a composition approach. Only the PAWS approach in [29] from the QoS category fulfils this requirement, the other approaches ignore this very important requirement.
- **Dynamic Aspect:** It is the most ignored area by the non-QoS-based approaches and is mostly covered by the QoS-based approaches. This is a very important requirement because it ensures that the composition schema is consistent and the faulty services can be replaced at run time.

- **Adaptability:** It is the process of the modification of an application before the problem occurs on the predefined strategies in the application. Adaptability is the most ignored requirement (as shown in Table 1) in all the composition approaches. Only the approach in [29], which is a QoS-based approach, fulfills this requirement.
- **Domain Independence:** Composition approaches should be generic enough to be applicable to different domains to be able to address different sets of problems. This requirement is covered by most of the approaches.
- **Correctness:** Is the guarantee to provide a certain kind of output under certain set of input and conditions. In this context, only two approaches; one which is the QoS-based approach in [31] and the other one, the non-QoS-based approach in [19], fulfill the requirements. The other remaining approaches fail to meet this requirement.

Table 1 shows that lack of scalability, non-determinism and adaptability are the major drawbacks of all the approaches. No approach without QoS support satisfies the requirements of dynamic aspect, non-determinism and adaptability, while no approach with QoS satisfies the requirements of scalability. Most of the approaches with QoS support are focused on the area of dynamic aspect while most of the approaches without QoS support focus on domain independence. A research that could complement the limitations of each approach categories is hence very much needed.

Table 1 shows that the PAWS [29] approach is the best approach among the rest because most of the parameters are satisfied by it. Semantic web approaches with QoS support are much better compared to the approaches without QoS support because it fulfills the maximum requirements that were identified in order to achieve successful composition of web services.

5 Conclusion

An overview of state of the art research in semantic web composition approaches was discussed in this paper. The existing works are classified into two categories; approaches with QoS support and approaches without QoS support. Six evaluation criteria were proposed for the purpose of systematically comparing the two approach categories to semantic web services composition; scalability, non-determinism, dynamic aspect, adaptability, domain independence, correctness, semantic capability and QoS awareness. Nevertheless, we do not claim that this classification is exhaustive. In each category, we give the introduction and comparison of selected approaches. This paper identified that semantic web composition approaches with QoS support are much better compared to the approaches without QoS support. The problems of the approaches without QoS support are the satisfaction of dynamic aspect, non-determinism and adaptability. On the other hand, the approaches with QoS support suffers from the scalability problem. An approach that complements each of the category is therefore needed to achieve successful web services composition.

References

1. Ran, S.: A model for web services discovery with QoS. *ACM SIGecom Exchanges* 4(1), 1–10 (2003)
2. Koehler, J., Srivastava, B.: Web service composition: Current solutions and open problems. In: Presented at the Proceedings of the ICAPS 2003 Workshop on Planning for Web Services, pp. 28–35 (2003)
3. Cabral, L., Domingue, J., Motta, E., Payne, T.R., Hakimpour, F.: Approaches to Semantic Web Services: an Overview and Comparisons. In: Bussler, C.J., Davies, J., Fensel, D., Studer, R. (eds.) *ESWS 2004*. LNCS, vol. 3053, pp. 225–239. Springer, Heidelberg (2004)
4. <http://uddi.org/pubs/uddi-v3.00-published-20020719.htm>
5. Benatallah, B., Dumas, M., Sheng, Q.Z., Ngu, A.H.H.: Declarative composition and peer-to-peer provisioning of dynamic Web services. In: *Proceedings of the 18th International Conference on Data Engineering 2002*, pp. 297–308 (2002)
6. Benatallah, B., Medjahed, B., Bouguettaya, A., Elmagarmid, A., Beard, J.: Composing and Maintaining Web-based Virtual Enterprises. In: *Workshop on Technologies for E-Services (2000)*
7. Casati, F., Ilnicki, S., Jin, L., Krishnamoorthy, V., Shan, M.-C.: Adaptive and Dynamic Service Composition in eFlow. In: Wangler, B., Bergman, L.D. (eds.) *CAiSE 2000*. LNCS, vol. 1789, pp. 13–31. Springer, Heidelberg (2000)
8. Lazcano, A., Alonso, G., Schuldt, H., Schuler, C.: The WISE approach to electronic commerce. *International Journal of Computer Systems Science & Engineering* (2000)
9. Muth, P., Wodtke, D., Weissenfels, J., Dittrich, A.K., Weikum, G.: From Centralized Workflow Specification to Distributed Workflow Execution. *J. Intell. Inf. Syst.* 10, 159–184 (1998)
10. Schuster, H., Georgakopoulos, D., Cichocki, A., Baker, D.: Modeling and Composing Service-Based and Reference Process-Based Multi-enterprise Processes. In: Wangler, B., Bergman, L.D. (eds.) *CAiSE 2000*. LNCS, vol. 1789, pp. 247–263. Springer, Heidelberg (2000)
11. Sandro Hawke, I.H., Prud'hommeaux, E., Swick, R.: *W3C Semantic Web Activity (2010)*
12. Berners-Lee, T.: *Services and Semantics: Web Architecture, W3C (April 2004)*
13. Bussler, C., Fensel, D., Maedche, A.: A conceptual architecture for semantic web enabled web services. *SIGMOD Rec.* 31:24-29 (2002)
14. Tsur, S., Abiteboul, S., Agrawal, R., Dayal, U., Klein, J., Weikum, G.: Are Web Services the Next Revolution in e-Commerce? (Panel). In: *Proceedings of the 27th International Conference on Very Large Data Bases*, pp. 614–617. Morgan Kaufmann Publishers Inc., San Francisco (2001)
15. Berners-Lee, T.: *Services and Semantics: Web Architecture, W3C (April 2004)*
16. Berners-Lee, T., et al.: *The Semantic Web*. *Scientific American* (May 2001)
17. Medjahed, B., Bouguettaya, A., Elmagarmid, A.: Composing Web services on the Semantic Web. *The VLDB Journal* 12, 333–351 (2003)
18. L_ecu_e, F., L_eger, A., Delteil, A.: DL Reasoning and AI Planning for Web Service Composition. In: *Web Intelligence*, pp. 445–453. IEEE, Los Alamitos (2008)
19. Rao, J., Dimitrov, D., Hofmann, P., Sadeh, N.M.: A mixed initiative approach to semantic web service discovery and composition: Sap's guided procedures framework. In: *ICWS*, pp. 401–410. IEEE Computer Society Press, Los Alamitos (2006)
20. Sohrabi, S., Prokoshyna, N., McIlraith, S.A.: Web Service Composition via the Customization of Golog Programs with User Preferences. In: Borgida, A.T., Chaudhri, V.K., Giorgini, P., Yu, E.S. (eds.) *Conceptual Modeling: Foundations and Applications*. LNCS, vol. 5600, pp. 319–334. Springer, Heidelberg (2009)

21. Aydın, O., Kesim Cicekli, N., Cicekli, I.: Automated web services composition with the event calculus. In: Artikis, A., O'Hare, G.M.P., Stathis, K., Vouros, G.A. (eds.) ESAW 2007. LNCS (LNAI), vol. 4995, pp. 142–157. Springer, Heidelberg (2008)
22. Dustdar, S., Schreiner, W.: A survey on web services composition. *Int. J. Web Grid Serv.* 1, 1–30 (2005)
23. Rao, J., Su, X.: A Survey of Automated Web Service Composition Methods. In: Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition, SWSWPC (2004)
24. Marconi, A., Pistore, M.: Synthesis and Composition of Web Services. In: Marco, B., Luca, P., Gianluigi, Z. (eds.) Formal Methods for Web Services, pp. 89–157. Springer, Heidelberg (2009)
25. Küster, U., Stern, M., König-Ries, B.: A classification of issues and approaches in automatic service composition. In: International Workshop WESC 2005 (2005)
26. Hull, R., Su, J.: Tools for composite web services: a short overview. *SIGMOD Rec.* 34, 86–95 (2005)
27. Milanovic, N., Malek, M.: Current solutions for Web service composition. In: Internet Computing, vol. 8, pp. 51–59. IEEE, Los Alamitos (2004)
28. Lécué, F., Silva, E., Pires, L.F.: A Framework for Dynamic Web Services Composition. In: Gschwind, T., Pautasso, C. (eds.) Emerging Web Services Technology, Birkhäuser Basel, vol. II, pp. 59–75 (2008)
29. Ardagna, D., Comuzzi, M., Mussi, E., Pernici, B., Plebani, P.: PAWS: A Framework for Executing Adaptive Web-Service Processes. In: Software, vol. 24, pp. 39–46. IEEE, Los Alamitos (2007)
30. Groenmo, R., Jaeger, M.C.: Model-driven semantic Web service composition. In: 12th Asia-Pacific Software Engineering Conference (APSEC 2005), p. 8 (2005)
31. Zeng, L., Ngu, A., Benatallah, B., Podorozhny, R., Lei, H.: Dynamic composition and optimization of Web services. *Distributed and Parallel Databases* 24, 45–72 (2008)
32. Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., Weerawarana, S.: Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI. In: Internet Computing, vol. 6, pp. 86–93. IEEE, Los Alamitos (2002)
33. Tsur, S., et al.: Are Web Services the Next Revolution in e-Commerce? (Panel). In: Presented at the Proceedings of the 27th International Conference on Very Large Data Bases (2001)
34. Arroyo, S., et al.: Semantic aspects of web services. In: Singh, M.P. (ed.) The Practical Handbook of Internet Computing, pp. 311–317. Chapman & Hall/CRC, Boca Raton (2005)
35. Ankolekar, A., et al.: Web Service Description for the Semantic Web. In: Presented at the Proceedings of the Semantic Web Conference (2002)
36. WSMO, Web Service Modeling Ontology–Standard (2005), <http://www.wsmo.org/2004/d2/>
37. Sivasubramanian, S.P., Ilavarasan, E., Vadelou, G.: Dynamic Web Service Composition: Challenges and techniques. In: International Conference on Intelligent Agent & Multi-Agent Systems (IAMA 2009), pp. 1–8 (2009)
38. Agarwal, V., Chafle, G., Mittal, S., Srivastava, B.: Understanding approaches for web service composition and execution. In: Proceedings of the 1st Bangalore Annual Compute Conference, pp. 1–8. ACM, Bangalore (2008)

Ontology Development for Programming Related Materials

Siti Noradlina Mat Using, Rohiza Ahmad, and Shakirah Mohd. Taib

CIS Department, Universiti Teknologi Petronas,
31750 Tronoh Perak, Malaysia

sitinoradlina@gmail.com, {rohiza_ahmad, shakita}@petronas.com.my

Abstract. As learning programming courses requires substantial self learning on top of the normal classroom lectures, supporting semantic search of programming related materials on the Web is seen to be one of the ways that can possibly enhance students' learning ability. Hence, in this research, an ontology of programming related materials is aimed to be developed. Once completed, the ontology will be embedded into an existing search engine to help students to do more effective search. However, as this research is a work in progress, in this paper, only the preliminary study and the development methodology of the ontology will be shared. The methodology involves rigorous participation of the domain experts and the results of evaluation by them are also presented in this paper.

Keywords: ontology, programming, semantic web, software development.

1 Introduction

Learning programming is not easy and according to Jeans Kaasboll in programming introductory courses, at the tertiary level (university level), the rate of failure and drop out in programming courses is high worldwide which is between 25% to 80% of the total enrolment [1]. Few problematic areas encountered by students when they study programming courses have been identified and among them as reported in [2] are:

- Hard to understand the syntax and some of the programming concepts.
- Cannot distinguish between programming knowledge and programming strategies.
- Hard to cope with the subject due to the characteristics of some novice programmers.

In short, we can say that it is hard to be good in programming because programming needs a continuous learning process in order to cope with the development of new technology [3]. Learning programming contains several activities like learning the language features, program design, and program comprehension. The normal approach in learning programming which is by listening to classroom lectures as well as reading textbooks is a good start to learn programming. However, studies have shown that it is also important to bring other aspects to the programming courses in order to enhance

students learning [2]. For example, students should take initiative to learn by themselves outside of the class period since it is hard or rather impossible for the lecturers to convey all information during the limited duration of the class time. The above suggestion is in line with the outcome-based education approach (OBE) which is implemented in some of the higher educational institutions all over the world. OBE rejects traditional focus on what is provided by the schools to their students, in favor of making the students demonstrate that they "know and are able to do" whatever the required learning outcomes are [4], [5]. Thus, self learning becomes more important to the students.

Studies in controlled environments suggest that the use of technology under the right circumstances will improve educational outcomes. Many educators believe that a new teaching method that incorporates technology is necessary to prepare students for work in the information age [6]. This has brought the Internet into the picture. Internet as many people know holds knowledge in abundance and one of the good sources for programming information. However, without proper searching system in place, not all of the intended information could be retrieved easily by the users. Hence, Semantic Web is one of the technologies currently preferred in solving this problem which is related to user search [3]. Ontology is the backbone or core components for Semantic Web. By using ontology, the semantic techniques are able to be achieved. It specifies the vocabularies, relationships of concepts, and plays a key role on the Semantic Web [7].

Thus, in this paper, the development process of an ontology that describes resources related to the programming domain as well as the semantic relationships between them will be presented.

2 Related Works

Semantic Web could be define as *"an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in co-operation"* [8]. Semantic Web enables machines to interpret data published in a machine-interpretable form on the Web to more meaningful content. It gives digital assistants and Web agents the capability to search the information which best correspond to the specific needs of a certain Web user [9]. Semantic is a study about meaning [10] where semantic of something is a meaning of something [11]. There might be more than one meaning for a word, so semantic is used to solve this problem of ambiguity by identifying the exact meaning of the word requested. Hence, semantic search is user search that uses data from the semantic networks to disambiguate queries so that more relevant results can be retrieved [11].

Ontology, on the other hand, is a formalization of semantics for a particular application domain. It specifies the definition of classes and relation modelling of domain objects and properties which are considered as meaningful for the application [12]. By formalizing and standardizing the meaning of words through concepts, ontology will enable a better communication between human and machine [13]. Common components of ontology are individuals (objects), classes, attributes and

restriction [14]. Few languages can be used for creating ontology like XML, RDF, RDFS and OWL. However, a widely accepted language to represent ontology is Web Ontology Language (OWL) managed by W3C [10].

Due to the increasing popularity of semantic-based searching, many ontologies have been constructed for many different domains. In other words, to semantically search for information which are related to a certain domain, the ontology for the domain will need to be constructed first. Hence, the success of the search will depend on the completeness of the ontology itself. For example, RightHealth [15] a site under Kosmix [16] is used for semantic search of health and medical domain. By using this website you can semantically search information regarding health and medical practices.

Currently, there are few popular semantic search engines that can be used to replace normal search engines such as Kosmix [16], Kngine [17], Hakia [18], duckDuckGo [19] Evri [20], Powerset [21], Semantifi [22] and others. However, these search engines are mostly focusing on a certain domain only and in the process of improving the other domains. For example, semantifi is currently focusing on finance, government, business and travel [22]. Meanwhile, Kosmix focuses more on social media [16]. Even though there are some search engines which might cover most of the domains, the completeness of the ontology for certain domains is still questionable [23]. For example when we search for terms from the programming domain using semantic search engine such as Hakia or Powerset, there are still some semantic features which are not covered by these search engines such as the list of possible relationships with other classes and the meaningful structure of the search results. Based on our research on existing search engines, we could say that there is still no semantic search engine available specifically for searching programming related materials from the web.

To develop a semantic search engine for programming resources domain, an ontology of programming resources must be developed first. Related ontology for programming resources has been studied and from our study we can say that for programming related ontology, there are very few ontologies that have been reported in the literature. Table 1 shows a list of those found and reviewed.

Of the several works above, only two of them are seen as most relevant to this research which are the C Programming Ontology [24] and Object-oriented Programming Ontology [25]. They basically present the inter and intra relationships of a course's content or concepts in the form of an ontology. However, they do not provide components in the ontology which talk about the resources of the information themselves. Thus, search results are not properly grouped based on their type of resources. Besides, some of the classes and attributes of those ontologies are not applicable to our domain of interest. Hence, they do not really support our research objective which is to create ontology of programming resources to help user search of programming related materials on the web.

Table 1. Ontology Reviewed

Ontology	Structure & method	Content
C Programming [24]	diagram, owl	Contain ontology representation for C Programming. The structure of ontology has been studied and applied in programming resources ontology during second draft of ontology.
Object-Oriented Programming [25]	owl	Contain ontology representation for object-oriented programming. The structure of ontology has been studied and applied in programming resources ontology during third draft (first version) of ontology.
Programming Knowledge Architecture [26]	diagram	Contain ontology representation on programming knowledge architecture. The ontology is simple and not really related.
Computer Science [27]	taxonomy hierarchy	Contain ontology representation on computer science. The ontology is simple and not really related.
Pizza [28]	owl	Contain ontology representation on pizza. This ontology provided as tutorial for protégé along with wine, newspaper, server and motor vehicle ontology. All this ontology are studied especially to understand more on how to use protégé.

3 Methodology

Table 2 below shows the stages of this research. There are 4 phase of the research and currently, this research is in its third phase which is ontology development and system development phase. In this paper, we will only discuss the methodology until phase 3. The methodology on system testing will not yet to be discussed.

Table 2. Project Methodology

Phase	Activity	Methodology
1	Data Gathering	Qualitative and Quantitative
2	Data Analysis	Qualitative and Quantitative
3	System Development	Ontology Development Methodology System Development Life Cycle
4	Testing	Qualitative and Quantitative

Figure 1 shows the system architecture of this research. The idea of this architecture was actually derived from [29]. Basically, the system architecture can be

divided into 2 components developments which are protégé ontology development and portal development. Ontology of programming related materials will be developed first and later it will be embedded into a search portal where search queries posted by users will be mapped / filtered semantically to the ontology. As mentioned in the abstract, since this research is still in progress, in this paper only the protégé development component will be presented and discussed in further detail.

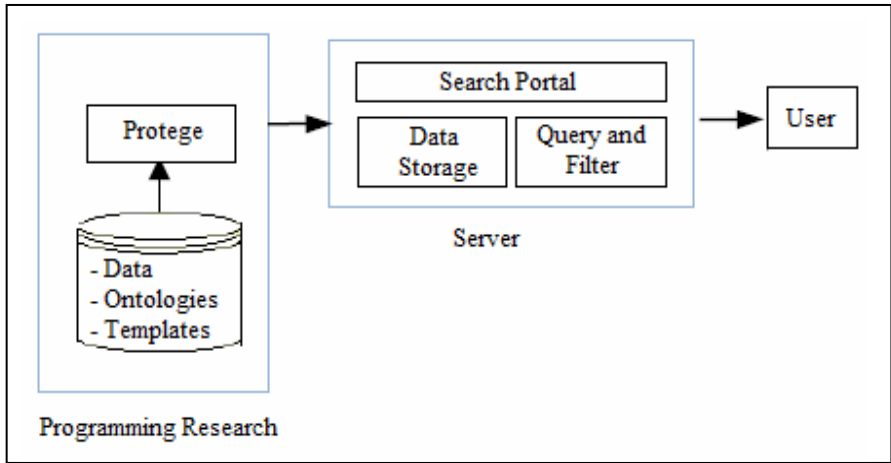


Fig. 1. System Architecture [29]

In order to construct the ontology for programming resources, the following approach has been taken. Firstly, preliminary data gathering and analysis was conducted. The activity can be divided into two types which were data gathering in term of literary study and data gathering in term of survey of potential users. Literary study basically involved the study of existing works on programming related ontology. This study served two purposes which were to find out the structure and ways for creating an ontology, and also to identify if any of the existing ontology can be integrated with the to be developed programming resources ontology. As for the user survey, 200 copies of pre-questionnaires have been distributed to some students and lecturers of programming courses at few local institutions such as Universiti Teknologi PETRONAS (UTP), Universiti Teknologi MARA (UiTM) and Kolej Politech MARA Bangi. The main purpose of this questionnaire was to support the need for such ontology and also to find out the weaknesses of the current search engines in searching information on programming materials.

Based on the results from the preliminary stage which is the structure of the existing ontology and way of creating ontology, the first draft of the ontology was constructed in the next stage. Even though there is no “correct” way or methodology for creating ontology [14],[30],[31], for this study, the iterative approach was followed. The steps involved in the approach were [14]:

Step 1: Determine the domain and scope of the ontology

- Domain: Programming Resources

Step 2: Consider reusing existing ontology

- First Cycle: There is no programming resources ontology available yet. However, existing ontology like C Programming ontology can be referred to get some basic ideas. During the first cycle, a basic draft for ontology of programming resources will be developed.
- Next Cycle: The draft of programming resources ontology will be used and enhanced

Step 3: Enumerate important terms in the ontology

- Example: Programming Language and Software.

Step 4: Define the classes and the class hierarchy

- There are few class and subclass involved in programming resources ontology such as Programming Entity and Database.

Step 5: Define the properties of classes-slots

Step 6: Define the facets of the slots

Step 7: Create instances

As stated in Step 2 above, the first draft will be enhanced for a better draft until it completes. For this, programming experts' opinions on the draft were gathered. 3 series of interviews were conducted with the experts in order to find the weaknesses and new ideas to improve the draft of the programming resources ontology. These experts can be divided into 3 categories which were:

1. Lecturers from IT Department who have had experienced in teaching programming courses
2. IT experts/staff from the industries
3. End users who are IT graduates

Currently, two of the interviews had been conducted. The first interview was conducted with a basic draft of programming resources ontology. The draft was improved to second draft after the analysis of the first interview's feedback. The second draft was used in the second interview for further improvement. Based on the results of the second interview, the complete draft called first version of ontology was constructed. This version will be validated and finalized by the third round interview. Interview approach was chosen throughout the ontology development and validation is made due to the suggestion by Lozano_Tello and Gomez-Perez [32],[33]. According to Lozano-Tello and Gomez-Perez [32], evaluation of ontology can be done by human where they will assess on how well the ontology will meets set of predefined standard, criteria and requirements [33]. The validation on the ontology of programming resources will be covered during third interview with IT experts. The purpose of this tests and evaluations is to make sure that the ontology of programming resources is updated to the current scenario, and it contains all related classes, relationship, properties and instances needed.

4 Result and Discussion

Results from the preliminary stage and the ontology construction stage are presented below.

4.1 Result from Pre-Survey

Figure 2 shows the result from pre-survey questionnaire.

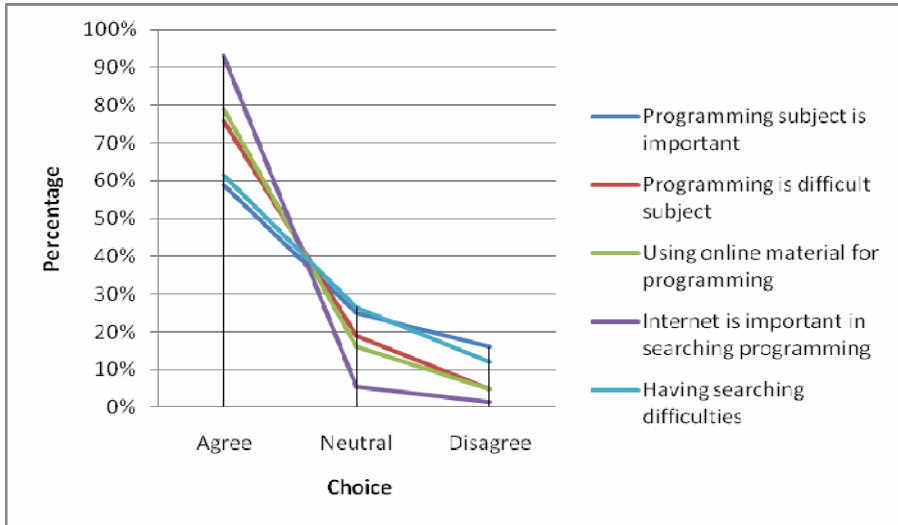


Fig. 2. Pre-survey Result

Based on data gathering and analysis done from pre-survey questionnaire, we can conclude that most of the respondents agreed that programming subject is important but it is also hard to learn. Most of them said that internet is important for programming materials searching and use online materials to help their study in programming. Based on the survey results, 79% respondents learn from online tutorials and references and 88% of them refer to online examples of codes, when doing programming.

The respondents said that it is easier for them to use the Internet as the source for finding information on programming as compared to other resources. This is because, internet facility is highly available to them and the monetary cost of searching is practically zero since Internet is supported by their organizations/institutions. However, as shown in figure 3, they also agreed that their searching skills and current Web structures have a few of weaknesses that can be improved such as:

- Hard to find wanted information and materials
- Difficulty in finding correct terms / keywords for searching information and materials.

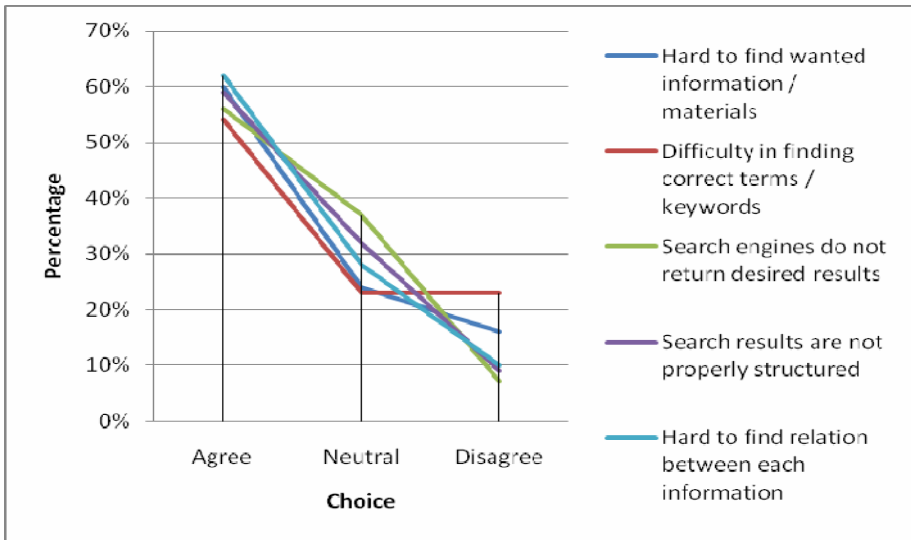


Fig. 3. Problem in Programming Searching

- The search engines do not return desired results
- Search results are not properly structured
- Hard to find relation between each information and materials on the Web

Based on the results above, it can be concluded that semantic web is useful to be implemented for searching programming resources over the Internet.

4.2 Data Gathering Result

After the usefulness of the idea was confirmed, a draft of the programming resources ontology was developed according to the preliminary study done on the existing semantic search engines and ontology related to this domain. Three series of interviews have been conducted to refine the drafted ontology of programming resources and validate it. Generally, the main purpose of these interviews was to obtain feedbacks from the experts on the drafted ontology. Specifically the first interview using first draft of ontology programming resources was to find out what are the suitable classes (and subclasses), relationship and instances for programming resources ontology. Based on the results of the interview, a second draft of the ontology was developed. A second interview was then conducted. This time the related terms and meaning for each class and instance were further explored. As a result, the completed draft called first version of the programming resources ontology was developed. The ontology was used as the focus of discussion in the third interview. This round of interview was more on finalizing and validating the ontology of the programming resources itself.

Based on the three interviews' results and preliminary studies, the programming resources ontology will contain six main classes which are:

1. Programming Entity
2. Database
3. Online Resources
4. Organization
5. Person
6. Tools

These six main classes are important classes when we discuss about ontology of programming resources. Programming Entity is of course important for the ontology since it is used as the basis for searching. It must be related to online resources such as book, journal and website. Furthermore, online resources are related to person. For example, the person who writes the textbook or journal. So this class is also important to have in this ontology. Similarly, the organizations who own the resources are also important. Programming domain is also related to database and tools. Programming subjects usually related to database subjects because some of the program might use data from a database. In other hand, tools are important for programming because without tools such as compiler a programming job is almost impossible to be done. Based on this reason, programming resources ontology also has these two classes as main classes. The rough idea of the main classes and relationships in the ontology of programming resources can be seen in figure 4 below.

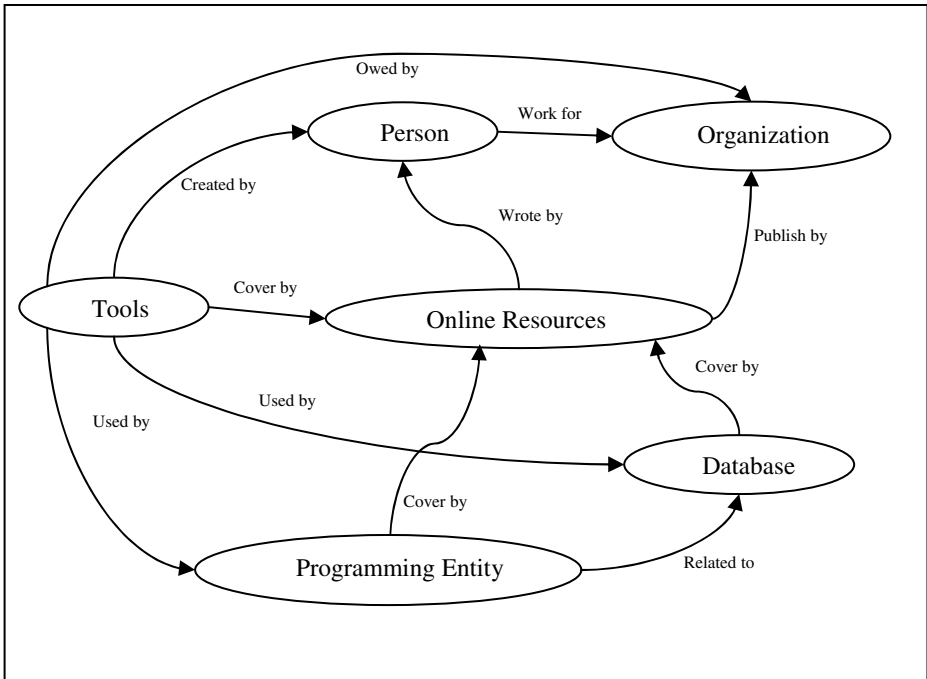


Fig. 4. A draft of Main Classes and Relationship

Table 3 below shows comparison of the ontology during the 3 stages of interviews.

Table 3. Comparison of Programming Resources Ontology during 3 Stages of Interviews

Session	1	2	3
Ontology Used	First Draft	Second Draft	First Version
No. of Classes	25	54	198
No. of Object Properties	12	30	48
No. of Data Type Properties	-	22	30
No. of Class with Instances	10	29	56
Existing Ontology Taken and Modified	-	C Programming Ontology	Object-oriented Prog. Ontology

During the first interview, a first draft of ontology shown in figure 5 below was used. This draft was created without embedding any existing ontology and basically has only the basic classes and subclasses of programming resources ontology. There were only 26 classes and subclasses in this draft and only 10 classes have instances. Besides that, there were no data type properties and only 12 object properties for the first draft.

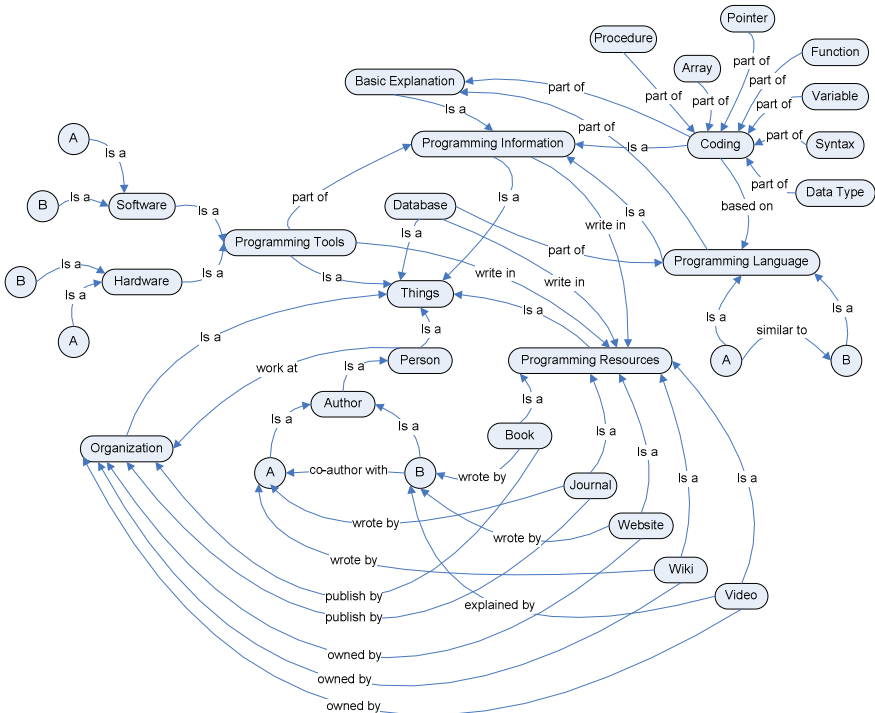


Fig. 5. The First Draft of Programming Resources Ontology [3]

Second draft of programming resources ontology as shown in figure 6 below was the result of the first interview and it was used during the second interview. The ontology has used some classes form C Programming ontology which is suitable to be embedded into the programming resources ontology. Basically it has 54 classes and subclasses, where only 29 of the classes and subclasses had instances. The second draft also had 30 object properties and 29 data type properties.

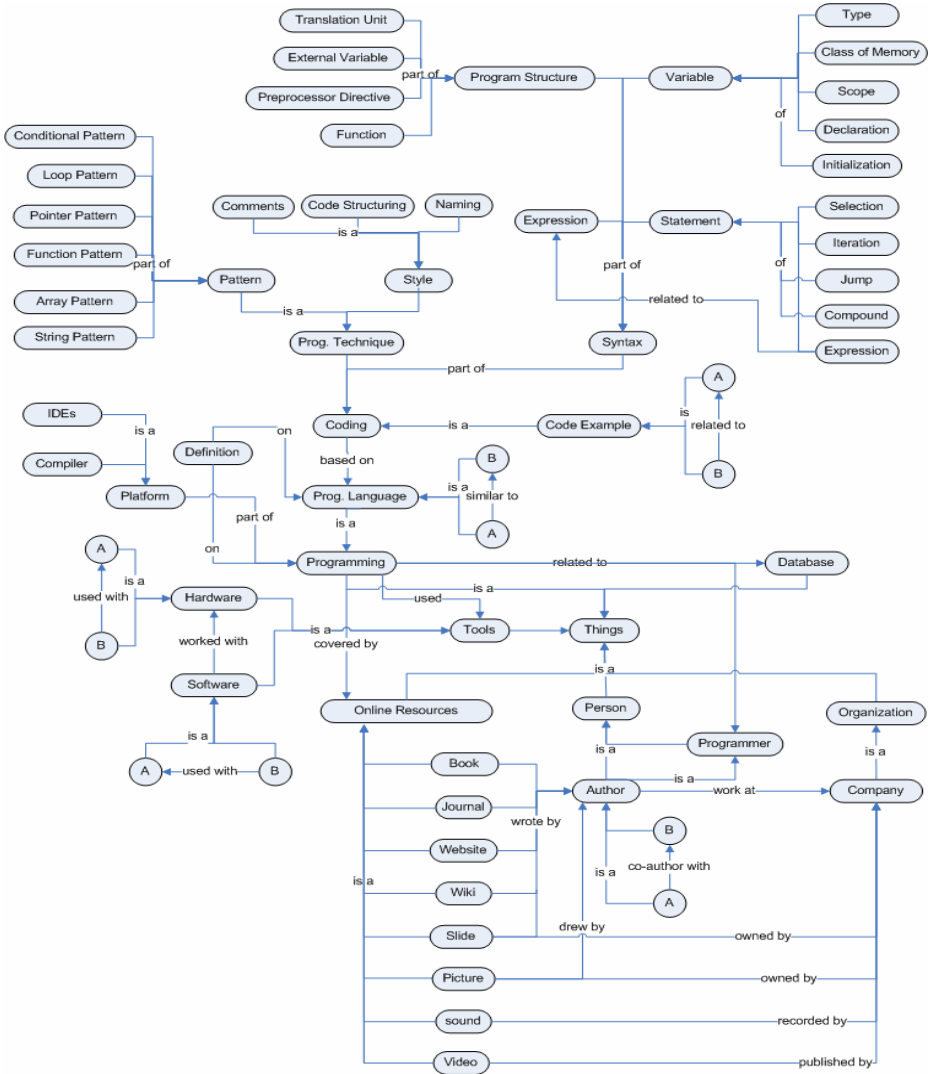


Fig. 6. The Second Draft of Programming Resources Ontology

From the second interview, third draft of programming resources ontology which is called first version of programming resources ontology was created. The ontology have use some classes form C Programming Ontology and Object-oriented Programming Ontology. Basically it has 198 classes and subclasses, and 56 of the classes and subclasses have instances. The ontology also has 48 object properties and 30 data type properties. The ontology will be validated by using human approach which at the current stage of the research has not been conducted yet.

The programming resources ontology was coded using Protégé application. Protégé is a free, open source application for knowledge-based framework and ontology editor [34]. Figure 7 shows a snippet of programming resources ontology in owl format created using protégé application. Meanwhile figure 8 shows a snippet of the abstract syntax form of programming resources ontology created by validator. These validators are owned by Matthew Horridge, University of Manchester [35], and University of Karlsruhe [36].

```

<rdfs:domain>
  <owl:Class>
    <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#Book"/>
      <owl:Class rdf:about="#Journal"/>
    </owl:unionOf>
  </owl:Class>
</rdfs:domain>
<rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
<owl:ObjectProperty rdf:ID="publish">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Author"/>
        <owl:Class rdf:about="#Organization"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
  <owl:inverseOf rdf:resource="#isPublishBy"/>
  <rdfs:range>

```

Fig. 7. A Snippet of Programming Resources owl

```
ObjectProperty(b:FROM)
ObjectProperty(b:TO)
ObjectProperty(a:calls Functional
domain(a:FunctionCall)
range(a:Function))
ObjectProperty(a:cite
inverseOf(a:isCitedBy)
domain(a:Journal)
range(a:Journal))
ObjectProperty(a:codedInLanguage
domain(a:Program)
range(a:ProgrammingLanguage))
ObjectProperty(a:contains)
ObjectProperty(a:cover
inverseOf(a:isCoveredBy)
domain(a:Online_Resources)
range(unionOf(a:ProgrammingEntity a:Database)))
ObjectProperty(a:draw
inverseOf(a:isDrewBy)
domain(a:Person)
range(a:Picture))
ObjectProperty(a:employ
inverseOf(a:workAt)
domain(a:Organization))
```

Fig. 8. A Snippet of Abstract Syntax Form of Programming Resources Ontology

5 Conclusion

As a conclusion, this research will help users especially students and lecturers to find information and programming resources on the Web. Semantic concept will be applied in this research and the project will focus on creating the ontology for searching information and programming related materials. Future direction of this project is to create a Website (or search engine) using the programming resources ontology and compared this Website with a normal Web to find which is more useful towards the users.

References

1. Kaasboll, J.: Learning Programming. INF-DID Informatics Didactics. Presentation slide, for Department of Informatics, University of Oslo (2002)
2. Ala-Mutka, K.: Problems in Learning and Teaching Programming: A literature Study for developing Visualizations in the Codewitz-Minerza Project. Codewitz Needs Analysis Report (2004)

3. Noradlina, S., Ahmad, R., Taib, S.M.: *Ontology of Programming Resource for Semantic Searching of Programming Related Material on the Web*. In: *International Symposium on Information Technology*, pp. 698–703 (2010)
4. Kasdirin, H.A.B.H.: *Introduction to Engineering Accreditation and OBE “The FKE UTeM Experience”*. Presentation slide, for Department of Control, Instrumentation, and Automation (CIA), University Teknikal Malaysia Melaka (2007)
5. Faculty of Electrical Engineering UITM Shah Alam: *Outcome Based Education*. In: *4th International Conference on University Learning and Teaching*, pp. 215–222 (2008)
6. Yasemin, G., Ismail, G.: *A Survey on ICT Usage and the Perception of Social Studies Teachers in Turkey*. *Journal on Education, Technology and Society* 11, 37–51 (2008)
7. Dou, D., McDermott, D., Qi, P.: *Ontology Translation on the Semantic Web*. *Journal on Data Semantic* 3360, 35–57 (2005)
8. Tim, B.-L., Hendler, J., Lassila, O.: *The Semantic Web*. *Scientific American: Feature Article* (2001)
9. Schahram, D., Dieter, F., Markus, L.: *The Realization of Semantic Web based E-commerce and its Impact on Business Consumers and the Economy*. Article for event “*Semantic Technologies – The Future of E-Business*”, Vienna (2006)
10. Alias, N.A.R., Noah, S.A., Abdullah, Z., et al.: *Application of Semantic Technology in Digital Library*. In: *International Symposium on Information Technology*, pp. 1514–1518 (2010)
11. Osman, N.A., Noah, S.A., Omar, N.: *Semantic Search in Digital Library*. In: *International Symposium on Information Technology*, pp. 1504–1507 (2010)
12. Rousset, M.-C.: *Small Can Be Beautiful in the Semantic Web*. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) *ISWC 2004*. LNCS, vol. 3298, pp. 6–16. Springer, Heidelberg (2004)
13. Sure, Y., Staab, S., Studer, R.: *Methodology for Development and Employment of Ontology Based Knowledge Management Applications*. Article published by *ACM Digital Library* 31, 18–23 (2002)
14. Natalya, F.N., Deborah, L.M.: *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880 (2001)
15. rightHealth Website, <http://www.righthealth.com/>
16. Kosmix Search Engine, <http://www.kosmix.com/>
17. Kngine Search Engine, <http://kngine.com/>
18. Hakia Search Engine, <http://www.hakia.com/>
19. duckDuckGo Search Engine, <http://duckduckgo.com/>
20. Evri Search Engine, <http://www.evri.com/>
21. Powerset Search Engine, <http://www.bing.com/>
22. Semantifi Search Engine, <http://www.semantifi.com>
23. Pandia Search Engine News,
<http://www.pandia.com/sew/1262-top-5-semantic-search-engines.html>
24. Sergey, S., Tatiana, G.: *Development of Educational Ontology for C-Programming*. *International Journal on “Information Theories and Application”* 13 303 (2005)
25. Michael, G.: *Object-oriented Programming Ontology*,
<http://www.professeurs.polymtl.ca/michel.gagnon/Ontologies/programming.owl>
26. Su, X., Zu, G., Liu, X.: *Presentation of Programming Domain Knowledge with Ontology*. In: *1st International Conference on Semantic, Knowledge and Grid (SKG)*, p. 131 (2005)

27. SHOE: Example Computer Science Ontology,
<http://www.cs.umd.edu/projects/plus/SHOE/cs.html>
28. Co-Ode Website: Pizza Ontology,
<http://www.co-ode.org/ontologies/pizza/2007/02/12/>
29. Sari, R.F., Ayunigtyas, N.: Implementation of Web Ontology and Semantic Application for Electronic Journal Citation System. *Journal of Emerging Technologies in Web Intelligence* 2, 34–41 (2010)
30. Zhanjun, L., Victor, R., Karthik, R.: A Methodology of Engineering Ontology Development for Information Retrieval. In: *International Conference on Engineering Design (ICED 2007)*, pp. 1–12 (2007)
31. Paea, P., Dejing, D., Gwen, A.F.: Ontology Database: A New Method for Semantic Modeling and an Application to Brainware Data. In: *20th International Conference of Scientific and Statistical Database Management (SSDBM 2008)*, pp. 313–330 (2008)
32. Lozano-Tello, A., Gomez-Perrez, A.: Ontometric: A Method to Choose the Appropriate Ontology. *Journal of Database Management* 15, 1–18 (2004)
33. Janez, B., Marko, G., Dunja, M.: A Survey of Ontology Evaluation Techniques. In: *Conference on Data Mining and Data Warehouses (SiKDD2005)*, pp. 166–170 (2005)
34. Protégé Website, <http://protege.stanford.edu/>
35. OWL Validator, <http://owl.cs.manchester.ac.uk/validator/>
36. Wonder Web OWL Ontology Validator,
<http://www.mygrid.org.uk/OWL/Validator>

User-Centered Evaluation for IR: Ranking Annotated Document Algorithms

Syarifah Bahiyah Rahayu¹, Shahrul Azman Noah¹,
and Andrianto Arfan Wardhana²

¹ Faculty of Information Science & Technology,
University Kebangsaan Malaysia, Bangi, Selangor, Malaysia

² Department of Information Technology
Chevron Indonesia Co.

Jakarta Pusat, Indonesia

sbrahayu@gmail.com, samn@ftsm.ukm.my, awardhana@chevron.com

Abstract. Since the introduction of Semantic Web, the practice of seeking and retrieving documents had been evolved. In this paper, the retrieved documents are ranked based on their annotated documents. We adopt two IR algorithms; Lucene Luke and ComFFICF. In order to verify the generated rankings, we run a user-centered evaluation, where it involved 10 human judges. Then, we assess the performance of ranking using NDCG metric. The assessment shows a ranking by ComFFICF algorithm outperforms a ranking by Lucene Luke. This method is proven to be one of preferable IR algorithms for searching and ranking annotated document.

Keywords: semantic web, annotation document, human ranking.

1 Introduction

Years ago, information retrieval (IR) applications are use to find documents based on the content of the document, not based on the knowledge of the document. Since the introduction of Semantic Web, the practice of seeking and retrieving documents had been evolved. Semantic web makes computer able to understand meaning of queries. With the support of semantic annotation and domain ontology, semantic web is able to assist people in querying rich documents. The query process is finding documents based on its annotation. These documents have been annotated, and the annotated version is known as document annotation. This document annotation is the knowledge about the content of the document itself. Semantic annotation represents a summary, a metadata, of the document based on a view, the domain ontology.

In this paper, we are using annotated document for the purpose of retrieving and ranking them based on some IR algorithms. We then investigate the results against human ranking. Our main objective is to evaluate the IR algorithms against human ranking. The paper is organized as follows. We discuss information retrieval and semantic web concepts in Section 2 and Section 3, respectively. In Section 4, we discuss the concept of user-centered evaluation. The evaluation of IR algorithms against human ranking is discussed in Section 5. The conclusion is given in Section 6.

2 Information Retrieval

Information retrieval (IR) is finding the most relevant information based on user's query. IR is defined as "finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)" [1]. Moreover, some IR systems are used to search structured information such as database system. The main factor in IR is to find relevant information that satisfies the user's preference. Over time, this discipline of science in searching information has been transformed from finding information on written document collections to finding information on digital document corpus. There are quite a number of IR systems such as web engines, library catalogues and scientific literature search. Traditionally, IR systems have been expected to provide relevant documents. However, in early information seeking, people tend to look for partially relevant documents [2]. Hence, a good IR system should provide relevant and partially relevant documents, which weigh them accordingly.

3 Semantic Web

The high demands of computers understand the meaning of vocabulary is a strong motivation for the emergence of Semantic Web. Semantic Web is a web with meaning, where, computers somehow understand the meaning of document and entities in that document on Web. According to Tim Berners-Lee et al, Semantic Web is "a web of data that can be processed directly and indirectly by machines." [3]. The World Wide Web Consortium (W3C) defines Semantic Web as a common framework that allows data to be shared and reused across application, enterprise, and community boundaries [4]. Hence, the framework applies some methods and technologies to allow and create machine-understandable representation of real-world entities and of relationships between entities [5]. The entities are uniquely identified and they have their own set of properties. Besides, properties can be used to relate two entities, so they have a relationship. Examples of methods and technologies that are formally expressed including Resource Description Framework (RDF), a variety of data interchange formats (e.g., RDF/XML, N3, Turtle, N-Triples), and notations such as RDF Schema (RDFS) and the Web Ontology Language (OWL). These methods and technologies provide a formal description of concepts, terms, and relationships within a given knowledge domain. Current Web is using HTML where it has insufficient description capabilities such as the followings:

- a. A web page does not understand the meaning of content
- b. Hyperlink is not describing the relationship between documents
- c. Problems in discovering information

Semantic Web is using ontology as a backbone of its application. Ontology is a representation of the real world entities. In addition, ontology is a formal representation of a set of concepts that defines the domain for each concept, and specifies the relationship between concepts. Since the process of creation of ontology is expensive and time-consuming, ontology mining tools have been developed. Some of the tools are including OntoMiner [6] and TaxaMiner [7] automatically construct

ontologies using bootstrapping, while Verity [8] automatically constructs a domain-specific taxonomy using thematic mapping.

Few de-facto standard ontologies are Gene Ontology (GO) and Medical Subject Headings (MeSH). These ontologies have been adopted by variety of Semantic Web project. There are few types of ontology languages such as RDFS and OWL. These languages are designed to annotate the documents as well as to enable semantic tagging of entities. RDF Schema (RDFS) was the first attempt towards developing an ontology language. RDFS was built upon RDF. It extends the RDF vocabulary with additional classes and properties such as `rdfs:Class` and `rdfs:subClassOf`. Latest ontology language is OWL, which extends RDFS components such as cardinality constraints, equality and disjoint classes. OWL language has three sublanguages, which are OWL-Lite, OWL-DL and OWL-Full.

Furthermore, RDF is a semantic data model to describe the objects (resources) and the relations among them using a triple syntax (subject-predicate-object) [9]. Therefore, the usage of RDF and OWL would allow more intelligent IR applications to be developed. The IR semantic search is taking advantage of the metadata associated with the entities to improve search quality. The relationships which defined in ontology would allow very complex queries to be answered.

4 User-Centered Evaluation: Human Ranking

One of the methods to verify the effectiveness of IR system is adopting a user-centered evaluation. In "NAGA"'s research work, they formulate 15 queries to compare their work with other IR systems and techniques [10]. There are 20 human judges to assess the top-ten results from all IR systems. Judges decide whether the result is highly relevant, correct but less relevant and irrelevant. Our approach to verify the ranking is similar to NAGA. In our experiment, we are using Roscoe's rule of thumb, where a sample size could be as small as 10 to 20 for a simple experiment with tight experimental controls [11]. Hence, there are 10 human judges to assess the ranking results. These results are categorized on the scale of relevant (2), less relevant (1) and not relevant (0).

In order to compute the cumulative gain, we are examining the retrieval results up to a fixed rank position. In this paper, we are using Normalized Discount Cumulated Gain (NDCG) metric to assess the performance of the ranking. First we calculate Discount Cumulated Gain (DCG) metric to measure the correlation between document rank and degree of relevance judgment. Normally, users tend to less likely examine later documents in the ranked result because it might be less valuable for them. These documents could be either of a relevant document or partially relevant document [12]. According to Järvelin and Kekäläinen, this metric DCG has the following benefits:

- a) the gain would weigh down as the document rank listed down
- b) the discount factor could be adjusted to model the user persistence in examining long ranked result lists [13, 14]

NDCG is an average metric that normalizes each entry in the DCG vector by the corresponding value in the ideal vector. In order to NDCG, it is essential to know

exactly relationship/correlation between the value of a relevant item decreases with a growing position in the result list, and relationship of every document with its relevance documents [15]. Besides, the later relevant document found in the ranked order would not influence the performance of NDCG. This recall basis metric is explicitly showing performance of IR techniques by giving the number of documents for which each NDCG value holds. However, it is difficult to apply NDCG to partial relevance feedback [16].

5 Evaluation

IR Algorithms. In this evaluation, we applied two IR algorithms into our research prototype retrieval engine, PicoDoc. The PicoDoc system is supported by a corpus that has pre-annotated documents as its data reference to run a query. In this experiment, a real-life dataset from news article corpus from websites of ABC and BBC is used. The corpus is based on OCAS2008 ontology [8], which is written in OWL. A guided query is used to place a query of a target instance. The PicoDoc system has an extendable framework built using Java language, to create and register semantic scoring method classes. For this study, we adopt two IR algorithms: Lucene Luke and ComFFICF. Table 1 below summarizes the list of IR algorithms:

Table 1. IR algorithms adopt in the IR system

Name	Concept Spread	Concept Weight	Normalized	Significance of Related Concept
Lucene Luke	Default Similarity	TF-IDF	Yes	n/a
ComFFICF	Related	FF-ICF	No	Fixed Weight (0.01)

a) Lucene Luke

Apache Lucene is an open-source indexer and searcher. Lucene uses a combination of Vector Space Model (VSM) and Boolean model for the retrieval. Basically, VSM is used to score the relevant documents and Boolean model is used to select the matching documents according to the user query. In addition, Lucene is a search tool kit which offers rich indexing and searching capabilities for web and desktop applications. This search kit is proven scalable and robust search applications [17, 18, 19]. Besides, Lucene has been employed into popular websites, for instance, Wikipedia, Source Forge, and Eclipse.

b) ComFFICF

This model is a modified version of FF-ICF (Feature Frequency–Inverse Concept Frequency) model [20]. In this paper, we adopt FF-ICF model into a weighting semantic annotation. In equation 1, consider a feature weighting factor wf extended from FF-ICF with a constant fixed weight, k that approximates the significance of a related concept. The total score, $totScore$ of target query, q in semantic documents, $Sdoc$ is a sum of all exact match, $ematch$ multiply with wf add the sum of the sum of feature match, $fmatch$ multiply with wf and k .

$$\begin{aligned}
\text{totalscore}(q, Sdoc) &= \sum_{i=1}^{|Sdoc|} (\text{ematch}(q, t_i) \times wf(t_i)) + \\
&\sum_{i=1}^{|Sdoc|} \sum_{j=1}^{|KGq|} (\text{fmatch}(t_i, tkgq_j) \times wf(tkgq_j) \times k)
\end{aligned} \tag{1}$$

Measurement For the experimentation, we formulate 15 queries, and use them to compare the two IR algorithms. For each question, the top results are shown to human judges. On average, every result is assessed by 10 human judges. For each result of each system, the judges had to decide on a scale from 2 to 0, whether the result is Relevant (2), Less relevant (1), or Not relevant (0).

When the query is placed from the guided query, for instance, *Australia*, the URI of Australia is derived from the knowledge base. Then, the query runs against the corpus and the knowledge base. The system populates the query knowledge graph, resulting in 1158 numbers of exact matching and related statements. Then, system runs each scoring methodology against each document annotation in the corpus, resulting in ranking of the documents in the corpus against the given query resource.

Lastly, we use Normalized Discounted Cumulative Gain (NDCG) [21] to assess the performance of the ranking. Hence, NDCG is not dependent on the number of results returned by the system for a given query. Furthermore, NDCG could exploit the rank and the weight of relevant results in the result list. This metric is intensively used in IR benchmarking.

Table 2. Results

IR Algorithms	#Q	#A	NDCG
Lucene Luke	15	59	78.41%
ComFFICF	15	162	93.22%

Result. The Table 2 shows the results of our evaluation. ComFFICF performs very well where the all the 15 queries returns results. This is due to performance of ComFFICF that returns both exact match and related articles. The query returns quite a number of answers when the instance is found in PicoDoc knowledge base and corpus. This is because ComFFICF make full use of knowledge map to retrieve all the exact match and related statements.

In contrast, Lucene Luke performs less well. Since Lucene Luke is based on TF-IDF algorithm, the length of the articles had influenced the ranking result. In few cases, it retrieves less result or no answer at all.

As shown in Table 2, ComFFICF algorithm (93.22%) ranking outperforms the Lucene Luke (78.41%). ComFFICF is using concept spreading to get the knowledge map. However, Lucene Luke is based on TF-IDF algorithm. The document length greatly influences a TF IDF score. In the case of semantic annotation, the annotation itself is already a concise summary of a document. A short annotation is always preferred by the TF IDF scoring, often outweighing the number of exact match found.

6 Conclusion

The user-centered evaluation is important to assess the search ranking. Since human is the user for IR system, it is important to get the feedback from them. The usage of NDCG for the assessment would reflect the preferable ranking. In this study, we are retrieving news article based on instance of annotated documents in a knowledge base and corpus.

In this study, we adopt two IR algorithms; Lucene Luke and ComFFICF into PicoDoc. In order to verify the generated rankings, we run a user-centered evaluation, where it involved 10 human judges. Then, we assess the performance of ranking using NDCG metric. The assessment shows a ranking by ComFFICF algorithm (93.22%) outperforms a ranking by Lucene Luke (78.41%). This method is proven to be one of preferable IR algorithms for searching and ranking annotated document.

Acknowledgments. The authors would like to thank the reviewers for their detailed, accurate and helpful comments.

References

1. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
2. Spink, A., Greisdorf, H., Bateman, J.: From Highly Relevant to Non-Relevant: Examining Different Regions of Relevance. *Information Processing and Management* 34(5), 599–622 (1998), http://www-staff.lboro.ac.uk/~lsas2/pubs/articles_page6.html (Accessed: 10 Feb.2009)
3. Berners-Lee, T., James, H., Ora, L.: The Semantic Web. *Scientific American Magazine* (2001)
4. World Wide Web Consortium (W3C), <http://www.w3.org/standards/semanticweb/>
5. Stoyanovich, J.: Search and Ranking in Semantically Rich Applications. Ph.D. Dissertation, Columbia University (2010)
6. Davulcu, H., Srinivas, V., Saravanakumar, N.: OntoMiner: Bootstrapping Ontologies from Overlapping Domain Specific Web sites (2004)
7. Kashyap, V., Ramakrishnan, C., Thomas, C., Sheth, A.: TaxaMiner: An Experimental Framework for Automated Taxonomy Bootstrapping. *International Journal of Web and Grid Services* 1(2) (2005)
8. Chung, C.Y., Lieu, R., Liu, J., Luk, A., Mao, J., Raghavan, P.: Thematic Mapping from Unstructured Documents to Taxonomies. In: *CIKM* (2002)
9. Kara, S.: An Ontology-Based Retrieval System Using Semantic Indexing. M.Sc. CompEng. Thesis, Middle East Technical University (2010)
10. Kasneci, G., Suchanek, F.M., Ifrim, G., Ramanath, M., Weikum, G.: NAGA: Searching and Ranking Knowledge, icde. In: 2008 IEEE 24th International Conference on Data Engineering (2008)
11. Roscoe, H.J.: A Cross-Sectional Test of the Effect and Conceptualization of Service Value. *J. of Services Mktg.* 11(6), 35–50 (1975); Sekaran, U.: *Research Methods for Business: A Skill-Building Approach*, 3 edn. John Wiley and Sons, Chichester (2000)

12. Järvelin, K., Kekäläinen, J.: IR Evaluation Methods for Retrieving Highly Relevant Documents. In: Belkin, N.J., Ingwersen, P., Leong, M.-K. (eds.) Proceedings of the 23rd ACM Sigir Conference on Research and Development of Information Retrieval, Athens, Greece, 2000, pp. 41–48. ACM Press, New York (2000)
13. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
14. Järvelin, K., Kekäläinen, J.: Cumulated Gain-Based Evaluation of IR Techniques. ACM Trans. Information Systems, pp. 422–446 (2002)
15. Typke, R., Veltkamp, R.C., Wiering, F.: A Measure for Evaluating Retrieval Techniques based on Partially Ordered Ground Truth Lists. In: ICME, 1793–1796 (2006)
16. NDCG,
http://en.wikipedia.org/wiki/Discounted_cumulative_gain#Normalized_DCG
17. Lucene Applications (2005),
<http://wiki.apache.org/lucene-java/PoweredBy>
18. Edgar, M., Maarten, d.R.: Deploying Lucene on the Grid. In: Proceedings SIGIR 2006 workshop on Open Source Information Retrieval, OSIR2006 (2006),
<http://en.scientificcommons.org/21615939>
19. Venkatachalam, L.: SSP Scalability of Stepping Stones and Pathways. M.Sc in Comp Sc. and App. Thesis, Faculty of the Virginia Polytechnic Institute and State University (2008)
20. Sartori, G., Gnoato, F., Mariani, I., Prioni, S., Lombardi, L.: Semantic Relevance, Domain Specificity and the Sensory/Functional. Theory of Category Specificity, *Neuropsychologia* 45, 966–976 (2007)
21. Dittrich, J.-P., Salles, M.A.V.: iDM: A Unified and Versatile Data. Model for Personal Dataspace Management. In: VLDB (2006)

Efficient Wireless Communications Schemes for Machine to Machine Communications

Ronny Yongho Kim

Kyungil University, School of Computer Engineering,
Gyeongsan, Gyeongbuk, 712-701 Korea
ronnykim@kiu.ac.kr

Abstract. Machine-to-Machine(M2M) communication is expected to be one of major communication methods in the future 5th generation wireless communications. In M2M communications, there are important requirements: extremely low power consumption of devices and mass device transmission. In order to meet the requirements, a novel snoop based relaying method in cellular M2M communications is proposed in this paper. The proposed scheme consists of two steps: 1. snooping group formation including group head selection and group members assignment based on link quality or location, 2 snooping and relaying packets to/from group members. By employing the proposed scheme, efficient M2M communication can be achieved without changing physical layer of wireless communication standards while meeting the requirements of extremely low power consumption and mass device transmission. We also propose a novel timer based group formation scheme considering M2M devices' mobility and show its advantages with simulation.

Keywords: Machine to Machine Communication, Snooping, Relay.

1 Introduction

Nowadays, more and more devices are connected to the communication network [1] and it is expected that Machine-to-Machine (M2M) communication will be one of typical form of communications in the future 5th generation wireless communications [2]. Under certain deployment environments, where large coverage is required such as fleet management, livestock or wild animal monitoring and so on, using cellular networks for M2M communication would be more beneficial than using short range communication such as wireless local area network (WLAN) or wireless personal area network (WPAN). In cellular communications, devices typically consume more power than short range communications due to large coverage area. One of important requirements in cellular M2M systems is extremely low power consumption [3]. Typical state transition diagram of mobile systems is shown in Fig. 1 [4]. Power saving operation can be performed either in active state or idle state in cellular system [5], [6]. In active state, power consumption of wireless device is mainly caused by uplink communication with a base station (BS).

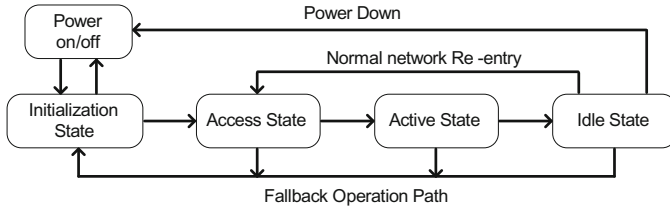


Fig. 1. State transition diagram in Mobile Station (MS)

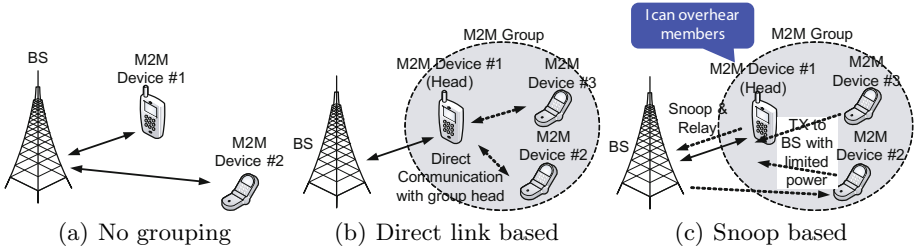


Fig. 2. M2M communication methods

M2M devices can communicate with BS in various ways as shown in Fig. 2. Cellular M2M devices can employ a similar communication method to human to human (H2H) communications: Fig. 2(a). Cellular M2M devices can communicate with the BS with the help of relay nodes, denoted as “group heads” in this paper: Fig. 2(b) and Fig. 2(c). Such communication methods using “group heads” can be called as group based communications which can address the power consumption problem in active mode. In the operation of the group based communications, a number of devices can be grouped together based on application type and/or geographical position in the cell and group head can relay packets to/from the BS on behalf of group member devices. The most efficient method to enable group based communication is to provide direct link between group members and group head as shown in Fig. 2(b). However, this requires new air interface standard with new radio frame structure. Typically it is anticipated that M2M system will be an add on feature to existing cellular systems such as WiMAX 1.0 system based on IEEE 802.16-2009 [7] or WiMAX 2.0 system based on IEEE 802.16m [5] which is selected as one of IMT-Advanced systems by ITU-R [8].

Thus, one of the efficient ways to enable group based communication in cellular networks without changing the physical structure of the existing air interface is through the proposed mechanism (Fig. 2(c)) in this paper in which a small number of “Group Head” nodes can lead the rest “Group Member” nodes for communication with the Base Station (BS) by employing a special technique named “Snooping”. The proposed scheme consists of two steps: 1. snooping group formation including group head selection and group members assignment

based on link quality or location, 2 snooping and relaying packets to/from group members. Since in the proposed scheme, M2M member devices employs small transmission power and resources allocated to the group can be reused by the other M2M groups in the cell, both extremely low power operation and mass device transmission requirements can be met.

The remaining part of the paper is organized as follows. In Section 2, system description of the proposed schemes is presented. Description on the proposed Snoop based M2M group communication scheme and Timer based group formation scheme is provided. In Section 3, performance of the proposed schemes is evaluated by analysis and simulation. Finally, we conclude the paper in Section 4.

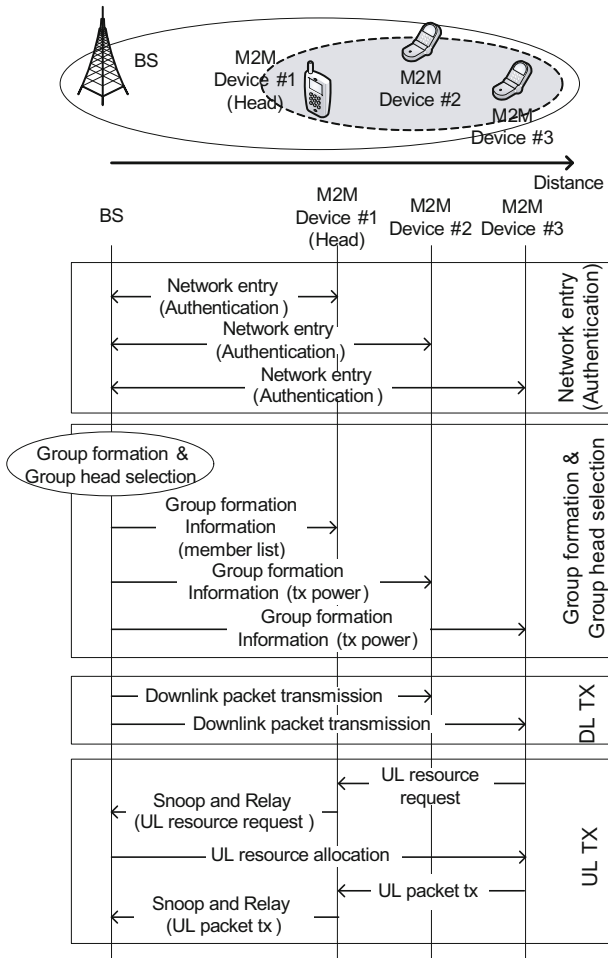


Fig. 3. Snoop based group communication procedures

2 System Description

Fig. 3 shows the message sequence of the proposed snoop based group communication procedures. Since cellular M2M systems typically use licensed bands, all M2M devices should be authenticated by the network during the “network entry phase” in order to access the cellular system. During the network entry, the BS adjusts the M2M devices physical parameters, such as transmission timing offset, transmission power and so on. With such physical parameter adjustment procedure, the BS is able to learn approximate geographical location of the M2M devices in the cell. Also the BS and the M2M devices negotiate their capabilities including supportable services and protocols. Therefore, after the network entry of the M2M devices, the BS is able to formulate M2M groups based on the application type and geographical location of M2M devices based on the knowledge acquired during the network entry procedure. Once group is initially formed by the BS, group formation information is delivered to the M2M devices. The group head receives member list through the group formation information message and the group member devices receive transmission power information through the group formation information message. M2M member devices do not need to be aware of which one is the group head device or which M2M group it belongs to because they will just transmit uplink packets to the BS with the notified power and the group head snoop the transmitted uplink packet and relay it to the BS. If there exists the already formed M2M groups and a new M2M device performs network entry, the BS makes the M2M device join the proper group by informing the group head of new member list and informing the M2M device of transmission power.

The time division duplex (TDD) radio frame structure of IEEE 802.16m is illustrated in Fig. 4(a). There are four frames, where one frame has downlink(DL) part and uplink (UL) part, in one superframe. Using one or more frames, a mobile station (MS) is able to receive or transmit packets as shown in Fig. 4(b). In order to receive packets from the BS, an MS has to decode all resource allocation control messages written in media access protocol (MAP) to identify if there is any packets to itself and the location of the packet. In case of UL transmission, the MS has to send UL bandwidth request through UL bandwidth request zone which can be identified through system information transmitted periodically. If the UL bandwidth request is successfully transmitted to the BS and the BS grants the MS's request, UL bandwidth is allocated through uplink MAP. The MS can transmit the packet using the allocated UL resource. In the proposed scheme, group members are allowed to access the DL packets from the BS directly since listening does not require much power consumption. However in case of UL, group member M2M devices transmit packets with limited power, which is notified by the BS, using the resource allocated by the BS. Since the group head is selected based on geographical position in the cell, the group head M2M device is able to snoop the packet transmission of group member devices and relay the snooped packets to the BS. Group member M2M devices do not know their packets do not reach the BS and are intercepted by the the group head M2M device. Since the resources allocated by the BS can be reused in the cell

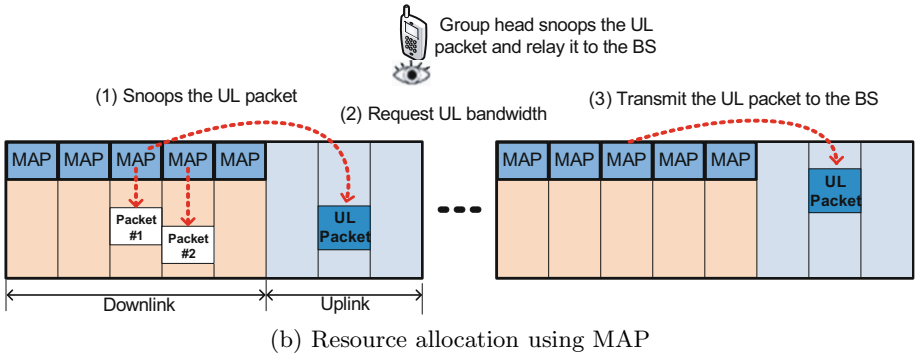
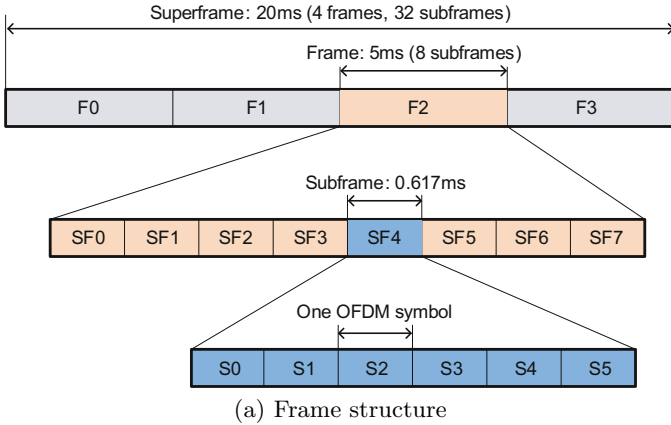


Fig. 4. Frame structure and resource allocation in IEEE 802.16m

due to limited transmission power, mass device transmission can be achieved by exploiting spacial multiplexing. Therefore, the proposed scheme improves the low power operation and capacity in the UL transmission. Since typical M2M communication is M2M clients report the measurement to the M2M server in the UL, the overall system performance improvement of M2M system through UL capacity improvement is huge.

When M2M group member devices have mobility, group formation scheme needs to be designed with consideration of the M2M group member devices' moving speed in order to prevent frequent group join and leave operation. M2M group formation problem is similar to the handover problem of Femtocell [10]. If fast moving M2M group member device joins a certain M2M group near by, it is very likely the M2M group member device soon leaves the M2M group. Therefore, we propose Timer Based Group Formation Scheme. The concept and flow chart of the proposed Timer Based Group Formation Scheme are shown in Fig. 5. The BS starts M2M_group_timer when the M2M member device enters the coverage of a certain M2M group. At the expiration of the M2M_Group_Timer,

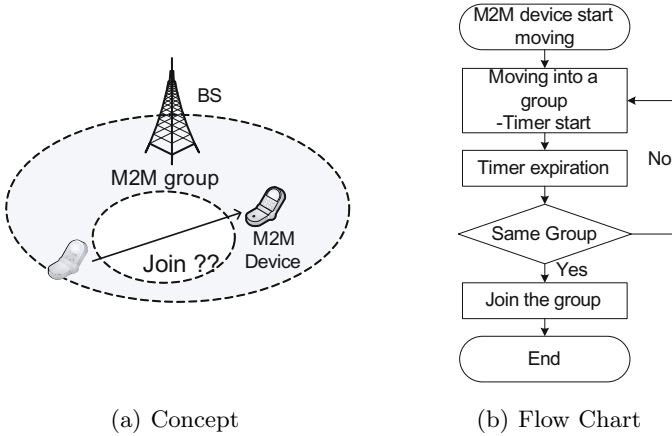


Fig. 5. Concept and flow chart of Timer Based Group Formation Scheme

Table 1. Pros and Cons of various communication methods for M2M.

Method	Pros	Cons
Direct	UL and DL capacity improvement Relatively small data relaying delay	Signaling overhead New radio frame structure
Snoop	Little signaling overhead for M2M device UL capacity improvement	Overhead on head M2M device Relatively large data relaying delay

if the M2M member device stays in the same M2M group, the BS assigns the MS to the M2M group. If the M2M member device moves out of the M2m group, the BS does not assign the M2M member device to the M2M group. Aforementioned procedure repeats while the M2M member devices are moving.

3 Performance Evaluation

In this section, performance comparison between two communication methods for M2M: the direct link based group communication, denoted as Direct, and the proposed snoop based M2M group communication, denoted as Snoop, is presented and also advantages of the proposed scheme is discussed.

3.1 Performance Comparison

As we can see from the Table 1, Direct shows the better performance in terms of DL and UL capacity improvement and it also shows better data transmission delay performance because the group head can autonomously allocate resources and perform error correction such as automatic repeat request (ARQ) and hybrid ARQ (HARQ) without the involvement of the BS. However, Direct requires extra signaling overhead of M2M devices to form M2M groups and it requires

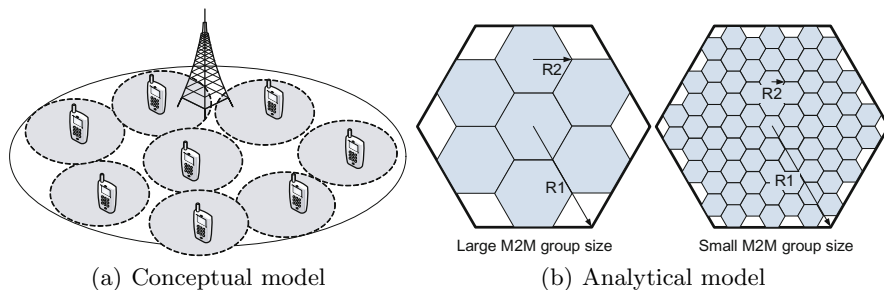


Fig. 6. Capacity increase with the proposed Snoop based group communication

new radio frame structure which has to be newly specified in the wireless communication standard. The proposed, Snoop introduces little signaling overhead for M2M devices since M2M group head takes care of group communication operation for its member M2M devices and provides UL capacity improvement. In Snoop, data transmission delay could be larger than Direct since the BS has to control all transmission including resource management and error control between the group head and members. However, in the typical operation of M2M communication, small delay (approximately, 5ms to 20ms) can be tolerable. Under certain M2M scenarios such as emergency situation, where even small delay can not be tolerable, M2M devices can directly communicate with the BS upon the happening of the urgent situation in Snoop.

3.2 Performance Analysis

The proposed snoop based M2M communication method has advantage of increasing UL capacity while providing low power operation to M2M member devices. Since M2M member devices transmit packets with power strong enough to just reach their M2M group head device, UL resource region allocated to M2M member devices of a certain M2M group can be reallocated to the other M2M groups. Through resource reuse of UL in a cell, UL capacity can be increased. The intuition of resource reuse among M2M groups is shown in Fig. 6(a). Since typical communication scenario of M2M is that M2M devices transmit very small UL packets to the network, by aggregating small packets of M2M members in a M2M group head huge overhead can be also removed leading to UL capacity improvement.

In order to show the benefits of resource reuse with M2M groups, analytical model shown in Fig. 6(b) is used. In order to show the benefits, M2M group numbers are given to M2M groups. A same group number is given to M2M groups which can reuse resources of other M2M groups which is similar to numbering cells considering reuse factor. Let us denote M2M group numbers as k , reuse factor as K , reuse distance as D , radius of a cellular cell as $R1$ and radius of a M2M group as $R2$. It is assumed that there are enough number of M2M devices

so that the BS can configure M2M groups freely and use same size for all M2M groups in the cell for simplicity. Fully symmetric hexagonal cell is also assumed. We can simply find the relationship between K and D [9] for M2M groups as follows:

$$\Delta = \frac{D}{R2} = \sqrt{3K} \quad (1)$$

where, Δ is the normalized reuse distance and reuse factor K can be written as the following form.

$$K = (i + j)^2 - i \times j \quad i, j = 0, 1, 2, 3, \dots \quad (2)$$

Possible values are $K = 1, 3, 4, 7, 9, 12, 13, \dots$. Because the lower the value of K is the lower the signal quality of reused channels, reasonable K should be employed.

When required signal to interference ratio (SIR) is γ_0 (dB), transmission power of M2M device is P (fixed transmission power is assumed for simplicity) and arbitrary propagation and antenna related constant is c , SIR (Γ) at a certain M2M device can be expressed as:

$$\begin{aligned} \Gamma &= (cP/R^4) / \left(\sum_k cP/D_k^4 \right) \\ &\simeq (cP/R^4) / (6cP/D^4 + 6cP/9D^4 + 6cP/16D^4 + \dots) \\ &= (D^4/R^4) \times (1/(6 \times (1 + 1/9 + 1/16 + 1/49 + \dots))) \\ &\simeq (1/7.4) \times \Delta^4 > 10^{\gamma_0/10} \end{aligned} \quad (3)$$

Reuse factor K for a required minimum SIR can be calculated.

$$K = \frac{1}{3} \times \left(\frac{D}{R} \right)^2 > \sqrt{0.82 \times 10^{\gamma_0/10}} \quad (4)$$

If a total number of channels is N , number of channels can be used in a M2M group with reuse factor K is $\eta = \lfloor N/K \rfloor$. If minimum required SIR is 20dB, $K = 9$ can be obtained.

Since number of channels that can be used in a M2M group is fixed depending on K , the smaller the M2M groups can be configured, the more M2M groups can be accommodated in a macro cell meaning more channels in a macro cell can exist. If the proposed Snoop based M2M group communication is employed, a macro cell use a total channel N . Whereas, by utilizing the proposed Snoop based M2M group communication, $N \times N_g$, where N_g is number of M2M groups in a macro cell, can be used in a macro cell. Using the examples shown in Fig. 6(b), UL capacity of ‘‘Large M2M group size’’ in a macro cell is approximately 7 times larger than the capacity of a macro cell only communication and UL capacity of ‘‘Small M2M group size’’ in a macro cell is approximately 63 times larger than the capacity of a macro cell only communication.

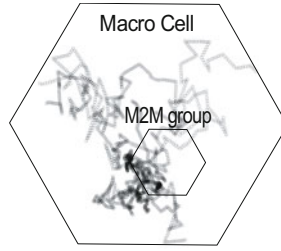


Fig. 7. Network topology for performance evaluation of Timer Based Group Formation Scheme

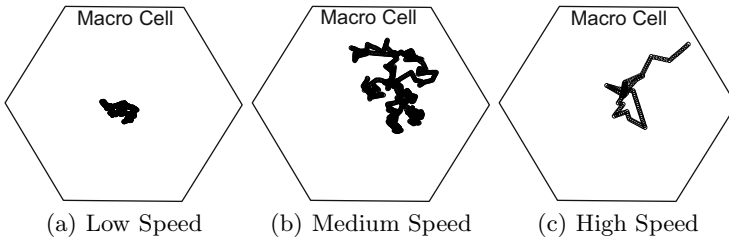


Fig. 8. Trace of MS's mobility simulated following the proposed mobility model

3.3 Performance of Timer Based Group Formation Scheme

Network topology to be used to show the benefits of the proposed, Timer Based Group Formation Scheme is shown in Fig. 7. Within a macro cell, M2M groups are located. For simplicity, one M2M group is shown in Fig. 7. It is assumed that there are enough number of M2M groups which can cover a coverage area of macro cell. However, in order to see the effects of M2M devices' mobility in relation to M2M group join and leave operation, i.e., sojourn time in a M2M group, when a M2M device moves out of one M2M group, the BS does not assign it to the other M2M groups. M2M device's movement is measured in every second and marked with circles. Fig. 7 shows the trace of 5 M2M devices moving at random speed for 1000 seconds.

We consider the motion of one M2M device around the coverage area of a cell randomly. Its initial speed (in km/h) and direction (in degrees) are generated with a uniform distribution of $U[10,80]$ and $U[0,360]$, respectively. The M2M device will change its speed and direction after a certain amount of time with an exponential distribution, with a mean value of 10 seconds. The new speed is uniformly generated with $U[10,80]$ if the current speed is below 10 km/h; otherwise, it is obtained using $U[v-10, v+10]$, where v is the current speed. The new direction is obtained from a Gaussian distribution with the mean as the current direction, and a standard deviation of 40 degrees. M2M devices are grouped into three classes: low mobile class (0km/h 30km/h), medium speed class (30km/h 80km/h) and high speed class (80km/h 150km/h). Fig. 8 shows

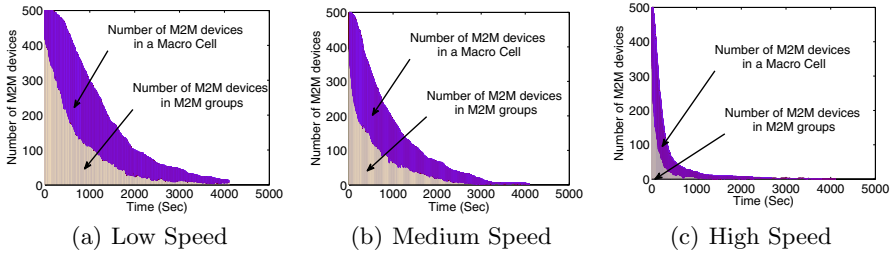


Fig. 9. Number of M2M devices in M2M groups and a macro cell. 500 M2M devices initiate mobility within a certain M2M group at time 0.

trace of an M2M devices' movement during 1000 seconds for different moving speeds: low, medium and high. As we can see from the results, slow moving M2M device is very likely to stay in the coverage area of a M2M group.

In order to see the distribution of M2M devices while moving within its coverage area of M2M group and macro cell, 500 M2M devices per mobility class is simulated. 500 M2M devices initiate movement within a certain M2M group at random at the same time and move around per the employed mobility model. As we can see from the results shown in Fig. 9, as speed increases, M2M device population decreases faster. Many M2M devices in the M2M group move out of the M2M group and finally moving out to other macro cells. The sojourn time of M2M devices in a M2M group gets smaller as M2M device's moving speed increases. Therefore, we can find some optimal value which can reduce unnecessary M2M group join and leave operation so that M2M devices can receive services for a long time without experiencing interruption caused by join and leave operation.

4 Conclusion

In this paper, novel efficient wireless communications schemes for M2M communications were proposed and discussed. More specifically, special scheme called "Snoop based M2M group communication" was proposed in order to meet the requirements of the extremely low power consumption requirement and mass device transmission requirement. The proposed scheme can be applied for various M2M scenarios without changing the physical layer of cellular wireless communication standard. We also proposed a novel timer based group formation scheme considering M2M devices mobility in order to prevent frequent M2M group join and leave operation. Through analysis and simulation, we showed advantages of the proposed schemes.

Acknowledgments. This study was supported by the Kyungil University Grant.

References

1. Morgan Stanley Research, The Mobile Internet Report: Ramping Faster than Desktop Internet, the Mobile Internet Will Be Bigger than Most Think (December 15, 2009)
2. IEEE 802.16ppc-10/0002r3, Future 802.16 network: Challenges and Possibilities (March 2010)
3. Machine to Machine (M2M) Communication Study Report, IEEE 802.16ppc-10/0002r7) (2010)
4. IEEE 802.16m-09/0034r4, IEEE 802.16m System Description Document (December 2010)
5. IEEE 802.16m Draft Amendment to IEEE Standard for Local and Metropolitan Area Networks, IEEE P802.16m/D7 (July 2010)
6. Kim, R.Y., Mohanty, S.: Advanced Power Management Techniques in Next Generation Wireless Networks. IEEE Communications Magazine (May 2010)
7. IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Broadband Wireless Access Systems, IEEE Std 802.16-2009 (Revision of IEEE Std 802.16-2004) (May 2009)
8. ITU-RM.2134, Requirements Related to Technical System Performance for IMT-Advanced Radio Interface(s) (IMT.TECH), draft new report (November 2008)
9. Zander, J., Kim, S.-L.: Radio Resource Management for Wireless Networks. Artech House (2001)
10. Kim, R.Y., Kwak, J.S., Etemad, K.: WiMAX Femtocell: Requirements, Challenges and Solutions. IEEE Communication Magazine 09, 55–59 (2009)

Efficient Data Transmission Scheme for Ubiquitous Healthcare Using Wireless Body Area Networks

Cecile Kateretse and Eui-Nam Huh

Internet Computing and Security Lab
Department of Computer Engineering, Kyung Hee University,
Seochon-dong, Giheung-gu, Yongin-si, Gyeonggi-do 446-701, Korea
{ckatere, johnhuh}@khu.ac.kr

Abstract. Data generated by wireless body area network (WBAN), are heterogeneous as they have different characteristics in terms of priority, transmission rate, required bandwidth, tolerable packet loss, delay demands etc; thus transmitting them in uniform fashion is unsuitable. This paper proposes an efficient data transmission scheme which classifies and prioritizes data patients according to their current status and their diseases; also we utilize multiple paths routing between the coordinator and gateway to provide a solution satisfying the delay requirements of different data types. Finally, it's shown that the proposed scheme is efficient for timely data transfer in WBAN.

Keywords: wireless body area network, quality of service (QOS), priority, multipath, end-to-end delay.

1 Introduction

A wireless body sensor area network (WBAN) is a network of multiple sensors attached to human body which can process, exchange sensed data, as well as communicate wirelessly to other in order to perform various tasks [1]. They have various applications including healthcare, lifecare and entrainments, where they monitor vital signs or human activities. Real-time data delivery is a challenge issue in those applications; as they deal with human data any delayed data delivery can endanger life as it can cause delayed response, particularly to emergency case. Moreover sensed data have different importance: they include critical packets and normal packets; this raise different requirement in quality of service.

In this paper we propose a quality of service (QOS) scheme for body sensor area network; a data classification and a prioritized forwarding on different routes are used to ensure proper QOS. We differentiate those critical packets from noncritical packets; in other words, we want to treat packets based on the content they carry.

Several routes exist between each coordinator and the gateway; the difference in packet content can be translated into the different choice of route. Several works for wireless body sensor treat all packets in the same fashion; thus we propose the development of new scheme in a real scenario to improve the QOS efficiency.

The rest of the paper is organized as follows. In section 2, we discuss the related works. In section 3, we present the proposed scheme. In section 4, we analyze the performance of the proposed scheme. Finally, in section 5, we conclude the paper and discuss future works.

2 Related Work

Various QOS methods have been studied for wireless body area network (WBAN): In [2] they propose an infrastructure for remote medical applications. In order to improve the delay and the transmission time of critical vital signals; a differentiated service based on priority scheduling and data compression is presented. In this model, a patient server receives instructions from a remote hospital server and configures the patient's WBAN accordingly. The wireless sensors transmit the signals to the server. Here, each type of vital signal receives a priority level and then is transmitted to the hospital server according to their priorities. In [3], they propose a QOS-aware routing service framework for biomedical sensor network where they prioritize routing service and user specific QOS metrics. In [4] a service prioritization and congestion control protocol is presented for wireless biomedical sensor networks in healthcare monitoring where they discriminate between different physiological signals and assign them different priorities.

However, in the above methods, they provide means to prioritize Electrocardiograph (ECG) and heart rate data above other patient data type due to its numerous parameters, the result of such prioritization is that other patient data type will endeavor delay in their transmission due to the dynamic change in patient status; thus we propose a dynamic priorities setting based on the patients requirements.

3 Proposed Scheme

The basic idea is to classify and prioritize the patient vital signs dynamically and also transmit data in multipath routing to ensure their timely delivery. In this section, we present the proposed scheme in details by first providing the system architecture, application scenario, and finally the data classification and prioritization.

3.1 System Architecture

The system architecture consists of groups of sensors attached to patient's body; a coordinator (CO), relays node (RN) and a gateway (GW) as illustrated in figure 1. Body sensors collect and send data to the coordinator which in turn follows data to a gateway (base station) through relays nodes if necessary. The gateway save patients data into the hospital database, where they can be accessed by hospital staffs in real time.

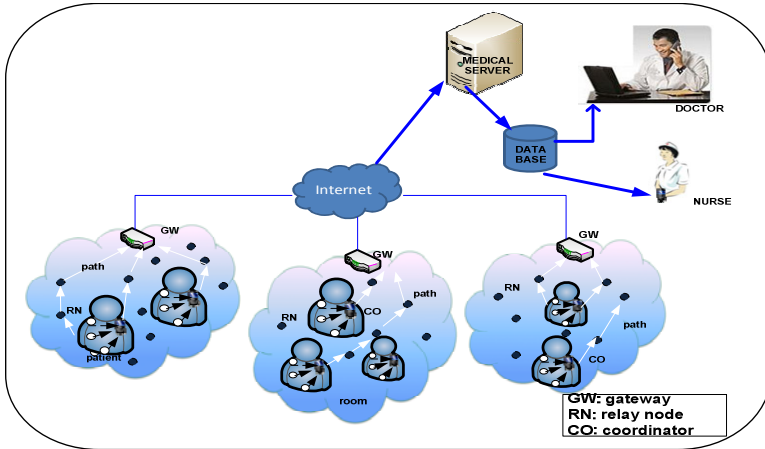


Fig. 1. System Architecture

3.2 Application Scenario

3.2.1 Scenario

We give firstly an application scenario which will be used throughout this work: we consider hospital environment where patients with various diseases are controlled employing ubiquitous sensor network to handle vital sign delivery including electrocardiography (ECG), heart rate, blood pressure, blood sugar, Saturation of Peripheral Oxygen (SPO2) and temperature. We mainly focus on three diseases such as sepsis [5], sleep apnea [6], intradialytic hypertension (IDH) [7], table 1 lists each monitored disease and corresponding vital signs or parameters.

Table 1. Monitored Diseases

diseases		sepsis	sleep apnea	IDH
Vital signs				
ECG		-	-	√
pulse		√	√	√
Blood pressure	Systolic	-	-	√
	diastolic	-	-	√
temperature		√	-	
SPO2		-	√	-
Breathing frequency		√	-	-

√ : corresponding vital sign;
 - : no corresponding vital sign.

3.2.2 Diseases Identification

In this section we describe shortly every disease; we also show diseases identification using tables 2 to 4;

In these, 'event type' means vital sign abnormal range, 'Criticality' means a level of emergency/risk.

a) Sepsis disease

Patient who suffer from sepsis disease become urgent, if it occurs two of the following criteria at the same time [5]:

- i. Temperature > 38.3°C or < 36°C
- ii. Pulse > 90 c/min
- iii. Breathing frequency > 20 c/min

Possible levels of criticality for sepsis patients are shown in table 2 where: normal level corresponds to normal sensed data; medium level is when there is one event type and the patient need care suspiciously; the high criticality level is mentioned for any two event type happened at the same time thus the patient requires immediate treatment.

Table 2. Disease Identification of Sepsis

Criticality level	Event type		
	Temperature > 38.3°C or < 36°C	pulse > 90 c/min	Breathing frequency > 20 c/min
Normal	no	no	no
Medium	yes	no	no
	no	yes	no
	no	no	yes
High	yes	yes	no
	no	yes	yes
	yes	no	yes
	yes	yes	yes

b) Sleep apnea

Patient who suffer from Sleep apnea become urgent, if it occurs two of the following criteria at the same time [6]:

- i. pulse rate < 60 beats/min or > 100 beats/min (for adult)
- ii. SPO2 < 90%

Possible level of criticality for sleep apnea patient are shown in table 3 where: normal level means that there is no event in sensed data; medium level is when there is one event type; the high criticality level is mentioned for two event type happened at the same time.

Table 3. Disease identification of sleep apnea

Criticality level	Event type	
	<i>pulse rate < 60 beats/min or > 100 beats/min</i>	<i>SPO2 < 90%</i>
Normal	no	no
Medium	yes	no
	no	yes
High	yes	Yes

c) Intradialytic hypertension (IDH)

Patient who suffer from IDH become urgent, if it occurs two of the following criteria at the same time [7]:

- i. Systolic blood pressure >20 mmHg
- ii. Decreasing mean arterial pressure >10mmHg

Possible level of criticality are shown in table 4 where level 1 refer to normal data ; medium level happen when there is one event type; the high criticality level is mentioned for two event type happened at the same time;

Table 4. Disease identification of intradialytic hypertension (IDH)

Criticality level	Event type	
	<i>Systolic blood pressure >20 mmHg</i>	<i>Decreasing mean arterial pressure >10 mmHg</i>
Normal	no	no
Medium	yes	no
	no	yes
High	yes	yes

3.3 Data Classification

In general, the sensed data observed by the sensors are transmitted to the gateway through a coordinator. But in WBAN, multiple events often occur; thus by specifying each node with corresponding role is necessary to schedule the events based on their criticality. In the following section the role of each node is described in details to handle important event properly.

3.3.1 The Role of the Source Node

When sensor node sense data, it immediately compare it with the defined threshold, and then decide if there is any event.

The event occurs when the data cross the threshold where the data type bit is filled as true, otherwise the data are normal and the data type bit is filled as false. This is illustrated in Algorithm 1.

After that data are delivered to the coordinator which performs data classification and sends it to the gateway through multipath referring on their criticality level.

Algorithm 1. event notification by the source node

Input:

- i. Threshold for vital sign x : thrshld_x
- ii. Data(s): current sensed data

Output: QoS requirements fulfilled

```

begin
01: for every period do
02:   for every vital sign  $x$  do
03:     if (data(s) >  $\text{thrshld}_x$ ) then
04:       category  $x \leftarrow$  event.
05:       Set the data type to true
06:     else
07:       category  $x \leftarrow$  normal
08:       Set data type to false
09:     end if
10:   end for
11: end for

```

3.3.2 The Role of Coordinator Node

The coordinator node has criticality table which is used to categorize patients based on the received data from the source nodes; data arrive as event or normal data, upon receiving the data the coordinator will use its criticality table to classify the data and take decision as follows:

If data come from source node, categorized as event, and if there is any corresponding event type stored before in the critical table, patient status will be taken as critical and data should be sent through shortest paths.

If data come from source node, categorized as event and there isn't any corresponding event type in the critical table, patient status is taken as moderate and data are sent in other paths. Also the coordinator will register this data as a new event in its critical table.

If data come from source node, categorized as normal, patient status is taken as normal and data are sent in remaining paths. The whole mechanism to classify and transmit events is described in figure 2. Every event notification has a unique ID. The uniqueness of an event notification is achieved by its timestamp.

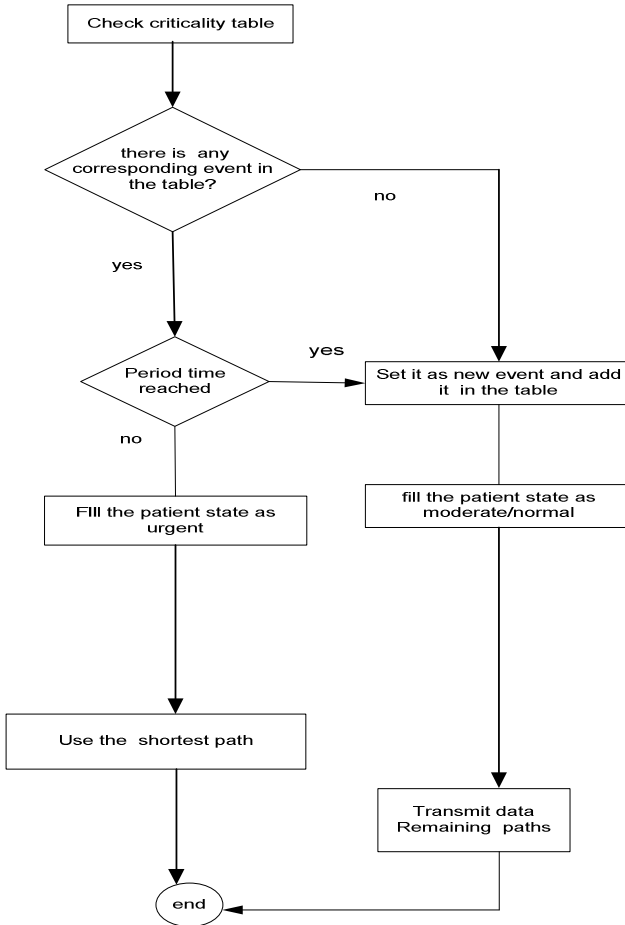


Fig. 2. Criticality level assignment

From this, we categorize patients as normal, moderate and urgent based on the sensed data and their criticality level (from table 2 to 4).

- **Normal:** A patient status is set to be normal when the coordinator node does not found any disease criteria/event.

- **Moderate:** A patient status is set to be moderate if the coordinator node perceives any one of the disease criteria; this means that the criticality level is medium.

- **Urgent:** A patient status is set to be urgent if the coordinator node finds two of the diseases criteria or all disease criteria simultaneously, this means that the criticality level is high. Algorithm 2 presents the patient categorization process in details.

Algorithm 2. Patient classification

- i. classification Input: data type
- ii. tbl_event: related data in the criticality table (1 or 0)
- iii. p:patient
- iv. time T:maximum time of event validity

Output: assign classes to patient based on type of the flow.

Begin

```

01: For every receive data from the source node do
02: while (time T not expired) do
03:   if(data type == event)&&( tbl_event==1)
04:     then
05:       Status  $p \leftarrow$  URGENT
06:     else
07:       if(data type== event)&&( tbl_event==0)
08:         then status  $p \leftarrow$  MODERATE
09:       else
10:         if(data type == no event) then
11:           status  $p \leftarrow$  NORMAL
12:         end if
13:       end if
14:     end while
15: end for

```

3.4 Data Transmission

To support real time data delivery, we adopt multiple paths routing, all body sensor send their data to the coordinator, which sends them to the gateway using multipath as illustrated in figure 3.

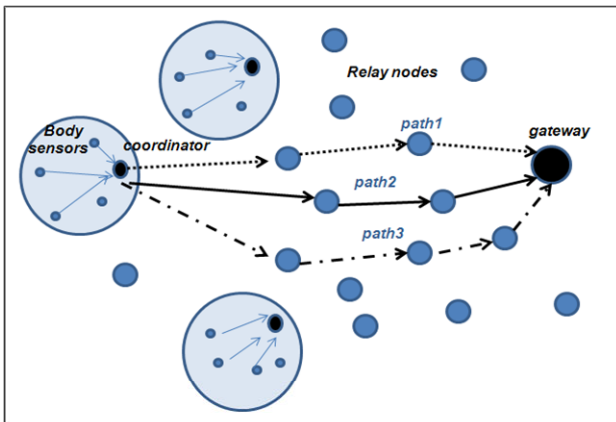


Fig. 3. An example of multipath routing for WBAN

The scheme consists of path discovery, path maintenance, traffic classification and path selection based on the assigned classes. We assume that multiple paths have already been established, and in this section we focus on path selection based on the data types.

We refer on class assigned in section 3.3 and give priority level on the data traffic; thus we use alternative routes for data transmission referring to data type, as shown in algorithm3.

We employ Bellman Ford shortest path algorithm, where we use hop count as metric; the number of hop is computed over all available paths and the path with least hop is selected among all to transfer higher priority data ; while moderate and low priority packets will be transferred through remaining paths keeping shortest path free for higher priority packets.

As the data are not gathering to a certain path, it guarantee better performance in term of timeliness as the low priority data utilize less occupied paths.

Algorithm 3. data transmission

Input:

- i. patient status
- ii. multiple paths

Output: assign path based on the patient status to ensure QOS

Begin

```

01: if (status x== URGENT) then
02:   set packet priority to 3
03:   else if (status x == MODERATE) then
04:     Set packet priority to 2.
05:   else if (status == NORMAL) then
06:     set packet priority to 1
07:     if Packet Priority=3 then
08:       forward the packet to the shortest path
09:     if Packet Priority is =2 then
10:       forward the packet to other paths
11:     if Packet Priority =1 then
12:       forward the packet to the remaining paths
13:     end if
14:   end if
15: end if
16: end if
17: end if

```

4 Performance Evaluation

In this section a theoretical evaluation is presented; where we analyze the end-to-end delay for different data traffic.

We consider the end- to- end delay for packet from the source node to the gateway and it's computed using equation (1), adopted from [8]:

$$e - 2 - e \text{ delay} = 2D \frac{1 + \left(\frac{f+r}{m}\right) + f/r}{|\mu| + \sqrt{\mu^2 + 2\tau(f+r)}} \tag{1}$$

Where:

- **D** is the distance from the source node to the gateway,
- **f** is the probability that the packet is lost,
- **r** is the time out of the packet,
- **m** is the waiting time before retransmitting data when there were lost,
- **μ** is the service rate, and **τ** is the arrival rate.

We consider that higher priority packets (with priority 3) are transmitted to the shortest path with probability P=1 while others packets are transmitted with equal probability 1/n in remaining paths, where n is the number of paths.

The parameters are D=10, m=0.1, τ=10 pkts, μ=50 pkts/s and μ=30 pkts/s respectively.

We compare the end-to-end delay of three traffic classes: critical, moderate and normal and the result are shown in figure 4 and figure 5.

We compute the end to end delay after the events occurs; the numerical result are plotted in Figure 4 which shows that the end- to- end delay of higher priority packets is lower than others kind of traffic.

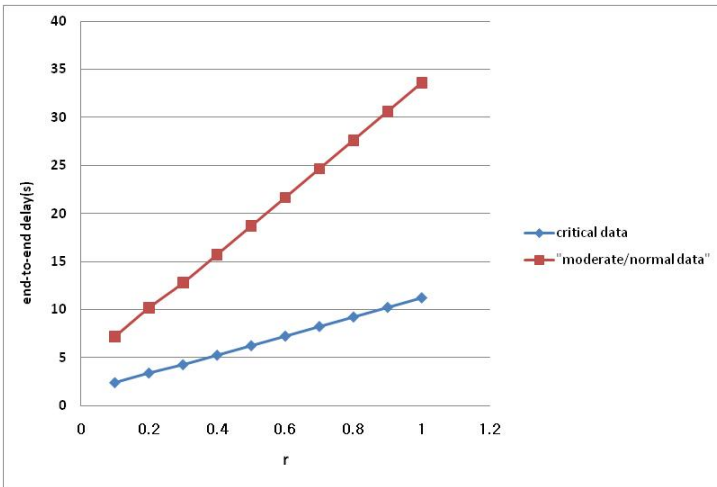


Fig. 4. End –to- end delay for different traffic

Next, we repeat the comparison with the same parameters setting as in figure 4 except the service rate μ, now is 30 pkts/s rather than 50 pkts/s.

The numerical result are plotted in Figure 5 which shows that the end-to-end delay of higher priority packets is still lower than normal and moderate traffic even if the service rate is changed.

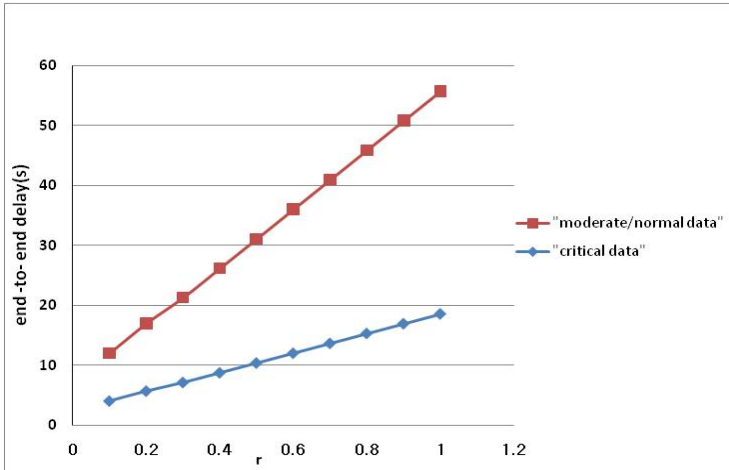


Fig. 5. End -to-end delay for different traffic

Above results shows that our scheme with differentiated service quality performs better than without differentiated service in relation to end-to-end delay on each traffic type.

5 Conclusion and Future Works

In this paper, an efficient data transmission scheme for WBAN based on QOS provisioning is presented. At the coordinator node the sensed vital sign are classified and assigned different priority based on the patient disease and its current status; thus the higher priority data are sent to shortest path and the low priority data are sent in the remaining paths, to ensure timely data delivery. The results show that our scheme can provide low end- to- end delay for critical data.

This work is still evolving and in the future work; we will consider other QOS parameters.

Acknowledgement

This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MEST) (No. 2010-0016959) and by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2011-(C1090-1121-0003)).

References

1. Yuce, M.R.: Implementation of wireless body area networks for healthcare systems. *Sensors and Actuators A physical*, 162 (2010)
2. She, H., Lu, Z.: network-based system architecture for remote medical applications. *Asia-Pacific Advanced Network, China* (2007)
3. Liang, X., Balasingham, I.: A QOS-aware routing service framework for biomedical sensor network. In: *IEEE ISWCS* (2007)
4. Hossien, M.: A Novel Congestion Control Protocol for Vital Signs Monitoring in Wireless Biomedical Sensor Networks. In: *IEEE WCNC* (2010)
5. Dessart, N., Fouchal, H., Hunel, P., Vidot, N.: Anomaly Detection with Wireless Sensor Networks. *IEEE NCA* (2010)
6. Suzuki, T., Kameyama, K.-i.: Development of a sleep apnea event detection method using photoplethysmography. In: *32 Annual International Conference of the IEEE EMBS* (2010)
7. Yung Cheng, W., Tsu-Yun, L.: A WSN- based wireless monitoring system for intradialytic hypertension of dialysis patients. In: *IEEE sensors* (2009)
8. Erol, G.: A diffusion model for packet travel time in a random multihop medium. *ACM transactions on sensors networks* 3 (2007)

An Algorithm to Detect Attacks in Mobile Ad Hoc Network

Radhika Saini and Manju Khari

Computer Science Department, Ambedkar Institute of Technology, New Delhi, India
myself_radhika@yahoo.co.in

Abstract. Each node in Mobile Ad Hoc Network (MANETs) communicates with each other to transfer the packet to destination node. Any anomalous behaviour of a node can confine it from executing this operation and even can disturb the whole network process. Therefore, the need of monitoring the nodes arises to keep a check on the behaviour of a node. In this paper, an algorithm is proposed to monitor the nodes & to check if a node is under attack or not. Moreover, a second layer of security is added which is furnished by a testbed to monitor the nodes.

Keywords: Mobile Ad Hoc Networks (MANETs), Nodes, Availability, Security, Attacks.

1 Introduction

Mobile Ad Hoc Network is a kind of network which does not require any fixed infrastructure (or central entity) to work. Such network can be formed within few minutes which consist of mobile nodes. An example of MANETs is bluetooth [1] where data is transferred between mobile nodes like cellular phones or laptops. In MANETs, each node works as a host as well as a router i.e. a node can transmit and receive the packets as a host and routes the packet to the destination as a router. Each node, before forwarding the packet to next node, decides the routing protocol [2] to route the packet.

A packet can reach the destination node within a single-hop or in multi-hop.

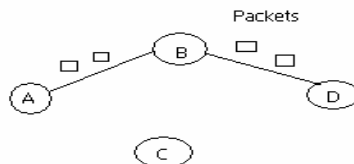


Fig. 1. Packet forwarding in Hops

For example in figure 1, A is the source and if destination is node B then packet reaches in single-hop and if destination is node C then packet reaches through

multiple hops. All the links between nodes are wireless. Military, emergency operations (like crowd control, search & rescue, commands operations) and collaborative computing are the fields where mobile ad hoc networks can be utilized.

MANETs has numerous security issues [3][4] like open wireless medium, shared radio broadcast channel, limited bandwidth, resource constraint and dynamic network topology which makes such network vulnerable to attacks [5] (eg. Blackhole attack, Wormhole attack, Rushing attack etc). These issues are exploited by intruder which breaches the security principles [6] (Confidentiality, Integrity, Availability, Authentication, Non-Repudiation) .Therefore it is mandatory to preserve all the security principles so that the entire network operation should not get disturbed. This paper is organized as follows: Section 2 presents the literature review on proposed methods for detecting nodes under attacks. Section 3 describes the design of algorithm proposed in this paper. In section 4, an algorithm is proposed which will detect the whether a node is under classified attack or unclassified attack or not under any attack. Conclusion and future work is given in Section 5.

2 Related Work

Every node in MANETs can take part in network operations if it is not under any attack. Any compromised node in the network can disturb the whole process. Therefore it is important to keep a check on the behaviour of a node to ensure that it is not under any attack. Numerous methods have been proposed to detect the status of nodes which are as follows:

- Intrusion Detection Systems (IDS) – Anomaly based IDS is mainly used in MANETs to detect any kind of intrusion in the network. Profiles are maintained in databases of IDS to match the anomaly. These profiles can be static or dynamic in nature. The problem with such system is that it is difficult to make a perfect profile. Moreover false alarm rate is higher [7].
- Random Walker Detectors (RWD) – This detector moves randomly from one node to other node to detect the node's activities. It monitors each node for a malicious behaviour and migrates to the selected node. This RWD has a specification based detection engine for comparing the behaviour of nodes [8].
- Watchdog – This method proposed the concept of a watchdog node which has high power and high transmission range than other ordinary nodes. This node watches and monitors the surrounding nodes. It keeps the node's data in its buffer and compares it after a new node receives it. Watchdog node is also called path rather [9].
- A method was proposed to detect the malicious nodes in the network by calculating the routes mainly the shortest route and re-routes the packet around them. This approach withstands the attacks in mobile ad hoc networks and based on routing protocols [10].

- Another method based on watchdog was introduced which was based on AODV (Ad-hoc On-demand Distance Vector) routing protocol. A credence based mechanism determines a node's credit standing [11].
- Security framework proposed for availability in which every node is monitoring its nearest neighboring nodes. The monitoring results are cross-validated after packet forwarding process. In this approach, each node has a valid certificate which indicates that this node is not under any attack [12].
- A model was proposed to check the status of a node in which an anomaly based IDS monitors the node in the network. When an anomaly is found, it simply checks its database classification for a match. If the anomaly matches then the attack handled otherwise it considered as a negligible anomaly [13]. This negligible anomaly can be an attack which is not defined in the classification. Such attacks are known as unclassified attacks which a classification fails to detect. Moreover this model can stuck in the loop of negligible anomaly in case of unclassified attack. Therefore it fails to detect such attacks. Moreover this model can stuck in the loop of negligible anomaly in case of unclassified attack. Therefore it fails to detect such attacks. This model has a single layer of security which is furnished by anomaly based IDS.

The motive of this paper is to provide a second layer of security to the state model proposed in [13]. This second layer security is furnished through a testbed which will simulate the negligible anomaly. After simulation, it will give result based on it. Result can be an attack or a negligible anomaly.

3 Design of Algorithm

Proposed algorithm design mainly has four parts- Anomaly based Intrusion Detection System (IDS) [7][13], a classification [13], a testbed [14][15] and a defence [13]. Anomaly based IDS monitors the node's activity constantly. If any anomaly (deviation) from normal behaviour is found in the network then IDS uses its own database (i.e. the classification) to match the abnormal behaviour. Matching is done with the help of a classification. If it is matched then that particular node is under attack and is not available for further use. And if it is not matched then the testbed simulates the behaviour to test whether that can be an unclassified attack which classification fails to match or a negligible anomaly. The unclassified attack can be a known or unknown attack.

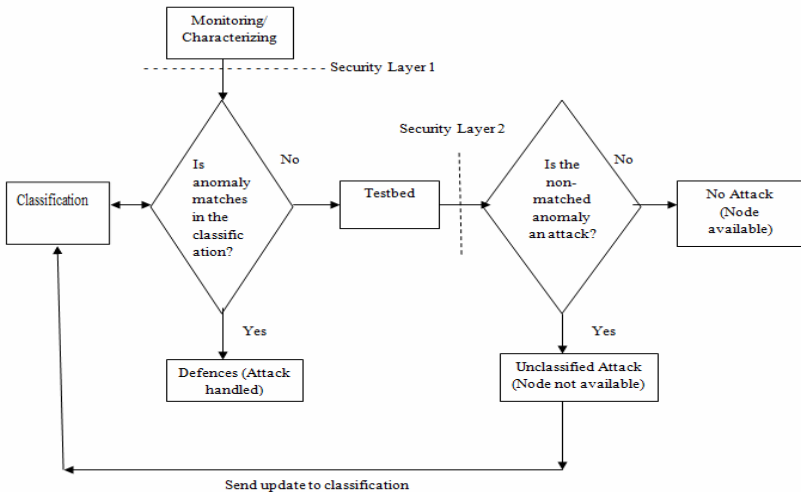
Proposed algorithm is an advanced version of the state model proposed by Bharat Bhargava, Ruy de Oliveira, Yu Zhang and Nwokedi C. Idika [13] in which a classification matches the anomaly with its stored abnormal behaviour and if it is not matched then it is returned as a negligible anomaly. This negligible anomaly can be an unclassified attack. To find such unclassified attack, this algorithm is proposed.

4 Proposed Algorithm

Definition of Unclassified attacks – These are the attacks which a classification fails to detect. Such attack can be an already known or a new attack. And that is why called the unclassified attacks.

In Ad Hoc Network, number of attacks [5] is defined through which an intruder attacks. An intruder always tries to find new and different ways to attack the network so that he cannot easily get caught. Therefore, there is a need of such a method which can easily detect the defined attacks as well as the other attacks. In this section, such an algorithm is proposed which provided two layer of security to the ad hoc networks – through IDS and then testbed. IDS will monitor nodes for any anomaly and testbed will simulate the unmatched anomaly.

Description of Algorithm – In proposed algorithm given below, each node is monitored with the help of anomaly based IDS. IDS already contain the abnormal (malicious) behaviour of a node in its database (present in classification). If any anomaly is detected by the IDS then the classification matches the malicious activity of a node with the stored activities in the database. If it matches then the attack is handled through the defined defences and if it does not match then the testbed simulates that particular behaviour of a node and determines whether that activity is an unclassified attack which can be a known or unknown attack or it is negligible anomaly which is a normal behaviour. The input to the testbed is the output from the classification in case of non-matched anomaly. Then this update is send to the IDS database. If a node is under any unclassified attack then that node will be considered unavailable for use and if a negligible anomaly is found then it is available for services to provide.



Proposed Algorithm

5 Conclusion and Future Work

The algorithm proposed in this paper will be able to detect all the classified and unclassified attacks in mobile ad hoc network. Unclassified attack can be an already existing attack which is known or can be a new attack (or unknown attack). The update mechanism to the IDS makes the data up-to-date which helps in finding the attack more easily. This algorithm will decrease the false alarm rate due to use of a testbed. Testbed will always simulate the non-matched behaviours to find the attacks and prevent the network from intruders.

Now the implementation of both the proposed work can be done on any simulator like NS2 or matlab and their results can be compared to know their respective efficiency. Designing a suitable testbed is a crucial task to do. Moreover time taken by both the methods and accuracy in detecting the attacks can be measured. Defining and setting the anomalous (or malicious) behaviour of a node is difficult.

Acknowledgments. I, Radhika Saini, would like to thank my college, Ambedkar Institute of Technology (situated at New Delhi) for providing me adequate resources to make this paper. I also would like to thank my guide Mrs. Manju Khari for her valuable suggestions and support.

References

1. Siva Ram Murthy, C., Manoj, B.S.: *Mobile Ad Hoc Networks-Architecture and Protocols*. Pearson Education, London (2004), ISBN 81-317-0688-5
2. Perkin, C.E.: *Ad Hoc Networking*. Pearson Education, London (January 2001)
3. Sheikhl, R., Chandee, M., Mishra, D.: *Security Issues in MANET: A Review*. IEEE (2010)
4. Rai, P., Singh, S.: *A Review of MANETs Security Aspects and Challenges*. IJCA Special Issue on Mobile Ad Hoc Networks (2010)
5. Wu, B., et al.: *A Survey of Attacks and Preventions in Mobile Ad Hoc Networks, Wireless/Mobile Network Security*, vol. 17. Springer, Heidelberg (2006)
6. Jangra, A., Goel, N., Priyanka, Bhati, K.: *Security Aspects in Mobile Ad Hoc Networks (MANETs): A Big Picture*. *International Journal of Electronics Engineering*, 189–196 (2010)
7. Sahu, S., Shandilya, S.K.: *A Comprehensive Survey On Intrusion Detection In Manet*. *International Journal of Information Technology and Knowledge Management* 2(2), 305–310 (2010)
8. Panos, C., Xenakis, C., Stavrakakis, I.: *IEEE Fellow - A Novel Intrusion Detection System for MANETs* (2009)
9. Patcha, A., Mishra, A.: *Collaborative Security Architecture for Black Hole Attack Prevention in Mobile Ad Hoc Networks*. IEEE, Los Alamitos (2003)
10. Mamatha, G., Sharma, S.: *A Highly Secured Approach against Attacks in MANETS*. *International Journal of Computer Theory and Engineering* 2(5), 1793–8201 (2010)
11. Jinghua, L., Peng, G., Yingqiang, Q., Gui, F.: *A Secure Routing Mechanism in AODV for Ad Hoc Networks*. IEEE, Los Alamitos (2007)

12. Jaisankar, N., Swamy, K.D.: A Novel Security Framework for Protecting Network Layer Operations in MANETs. *International Journal of Engineering and Technology* 1(5) (2009)
13. Bhargava, B., Oliveiral, R., Zhang, Y., Idika, N.: Addressing Collaborative Attacks and Defense in Ad Hoc Wireless Networks. In: *IEEE International Conference on Distributed Computing Systems Workshops* (2009)
14. Barolli, L., Ikeda, M., Xhafa, F., Duresi, A.: A testbed for MANETs: Implementation, Experiences and Learned Lessons. *IEEE, Los Alamitos* (2010)
15. Li, L., Zhang, H.: Research on Designing and Implementing an Experimental MANET Testbed. In: *IEEE International Conference on Communication Software and networks* (2009)

Integrated Solution Scheme with One-Time Key Diameter Message Authentication Framework for Proxy Mobile IPv6

Md. Mahedi Hassan and Poo Kuan Hoong

Faculty of Information Technology,
Multimedia University, 63100, Cyberjaya, Malaysia
{md.mahedihassan08, khpoo}@mmu.edu.my

Abstract. Proxy Mobile IPv6 (PMIPv6) is an effective mobility management protocol for next generation wireless networks which improves ubiquitous network access. However, PMIPv6 still suffers from lengthy handover latency and packet loss during the handover when Mobile Host moves to a new network. In order to improve the performance of PMIPv6, we proposed an integrated solution scheme with Media Independent Handover (MIH) and neighbor discovery message of IPv6 to reduce handover latency and packet loss. The proposed protocol does not have method to prevent from security threats such as replay attack and key exposure when mobile host first enters in PMIPv6 domain. In order to address this problem, we proposed one-time key with Diameter Message authentication framework which is based on the one-time key generation authentication protocol. It is expected the proposed framework is able to enhance security as well as reduce authentication latency.

Keywords: Proxy Mobile IPv6, authentication method, security analysis.

1 Introduction

The Proxy Mobile IPv6 (PMIPv6) is designed to provide an effective network-based mobility management protocol for next generation wireless networks that supports to a Mobile Host (MH) in a topologically domain [1] [2]. PMIPv6 extends MIPv6 signaling messages and reuse the functionality of Home Agent (HA) to support mobility for MH without host involvement. In the network, mobility entities are introduced to track the movement of MH, initiate mobility signaling on behalf of MH and setup the routing state required. The core functional entities in PMIPv6 are the Mobile Access Gateway (MAG) and Local Mobility Anchor (LMA). The main role of the MAG is to perform the detection of the MH movement and initiate mobility-related signaling with the MH's LMA on behalf of the MH. In addition, the MAG establishes a tunnel with the LMA for forwarding the data packets destined to MH and emulate the MH's home network on the access network for each MH. The main role of the LMA is to manage the location of a MH while it moves around within a PMIPv6 domain and it also includes a binding cache entry for each currently registered MH and also allocates a Home Network Prefix (HNP) to a MH.

With regard to authentication, when the MH first enters in the PMIPv6 domain, it sends Router Solicitation (RS) message to MAG. When MAG in the access network receives the request from the MH, the access authentication and authorization procedures are performed using a MH's identify before providing PMIPv6 services. While access is authenticated or network attachment events are notified, the MAG obtains the MH profile which contains MH-Identifier and uses it to access the MH's policy server (e.g. authentication, authorization and accounting [AAA] server), supports address configuration mode and retrieves the address of the LMA that serves as the MH's HA. After successful access authentication, MAG configures a proxy care-of-address (PCoA) for the MH and sends a proxy binding update (PBU) message including the MH-Identifier to the MH's LMA on behalf of the MH. In return, the LMA updates its binding cache entry (BCE) for that MH and checks policy store to ensure that the sender is authorized to send the PBU message. If the sender is a trusted MAG, the LMA accepts the PBU message and replies with a Proxy Binding Acknowledgment (PBA) that contains the MH's home network prefix assigned by the LMA. Upon receiving the PBA, the MAG establishes a bidirectional tunnel between its proxy CoA (PCoA) and the LMA address. Then, the MAG periodically sends Router Advertisement (RA) messages to the MH on the access link advertising the MHs home network prefix as the hosted on-link prefix. In a nutshell, in order to reduce the handover latency and packet loss, our proposed integrated solution architecture of PMIPv6-MIH is shown in fig. 1.

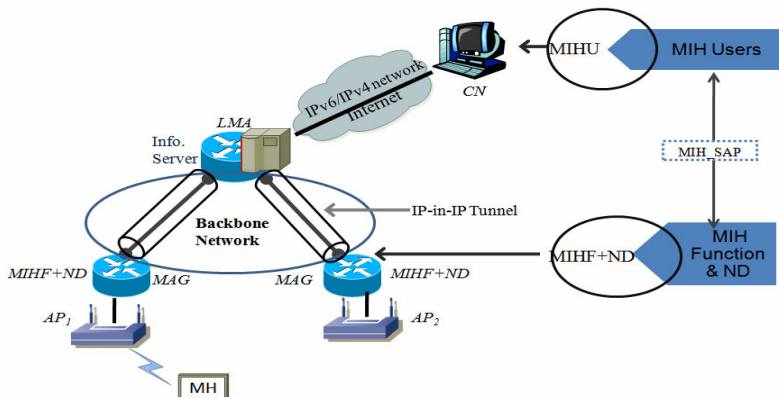


Fig. 1. Integrated solution architecture of PMIPv6-MIH

In our proposed integrated solution, it includes Media Independent Handover (MIH) and Neighbor Discovery (ND) messages. The key functionality is provided by MIH which is communication among the various wireless layers and the IP layer. The working group of IEEE 802.21 introduces a Media Independent Handover Function (MIHF) that is located in the protocol stack between the lower layer wireless access technologies and IP at upper layer. It also provides the services to the layer 3 and layer 2 through well defined Service Access Points (SAPs) [3].

Neighbor Discovery (ND) is a set of ICMPv6 messages and processes that determine the relationship by sending network information to the neighbor MAG before handover that can help to eliminate the need for MAG to acquire the MH-profile from the policy server/AAA whenever a MH performs handover between two Access Points (AP). It avoids the packet loss of on-the-fly packet which is routed between the LMA and previous MAG. This network information could include information about MH-profile which contains the MH-Identifier, MH home network prefix, LMA address (LMAA), MIH handover messages etc. The module of ND is used to provide the layer 3 movement detection. In the network, AP sends RA messages periodically to inform the MH about the network prefix. The prefix is the address of the AP. If RA messages contain a new prefix and inform the interface manager, MH receives these RA messages and determines where ND agent located. A timer is associated with the lifetime of the prefix. The prefix expired and a notification is sent to the interface manager when the MH loses its connection with the AP. The implementation of ND Agent is based on the information of ND for IPv6 which is provided by RFC 2461[4].

The objective of this paper is to propose an integrated scheme with one-time key Diameter Message authentication framework that is able to reduce authentication latency as well as prevent security threats such as replay attack and key exposure for PMIPv6-MIH. The rest of this paper is organized as follows: Section 2 presents related works, while Section 3 briefly explains the proposed authentication method for PMIPv6-MIH. Section 4 conducts security and performance analysis of the proposed authentication method. Finally, Section 5 concludes the paper and provides future works.

2 Related Works

To establish, update and tear down routes for mobility signaling messages of a MH, PMIPv6 is executed on the interface between a MAG and an LMA. However, there are many security threats to PMIPv6 that includes man-in-the-middle attacks such as intercept, flaw, modify, or drop such traffic, or denial-of service attacks on high-profile web servers such as banks, credit card payment gateways, and even root name servers, or redirect it to destination in collusion with the attacker with compromise or impersonation of a legitimate MAG or a legitimate LMA [5]. A compromised MH can also attack the PMIPv6 system. Through inspection, attacker can catch authentication data for MH and also spoofing attack can be done to MH's home network.

The current authentication problems on PMIPv6 can be summarized as follows:

- There is no way to authenticate the legality of a MH
- Compromise or impersonation of a legitimate MAG or a legitimate LMA
- Compromise or impersonation of a legitimate MH

In order to solve these problems, there are two commonly used authentication protocols implemented to secure authenticate of MH such as One Time Password and One Time Key Generation. One Time Key Generation is one part of One Time Password (OTP) because it used a time-synchronization type OTP function to

generate a key. Using the key, MH can authenticate when MH first enters in a PMIPv6 domain. When MH moves one network to another within same domain, MH accesses the new network to use that key.

2.1 One Time Password

An attacker can easily capture or stolen or attempts to crack traditional or static passwords. To overcome these problems network working group developed One-Time Password (OTP) system that is valid for only one login session. Based on some specific values, OTP generates temporary password that can be used only one time [6].

There are three approaches to generate password in OTP system. First approach: using a mathematical algorithm, OTP generates new password based on the previous password. Second approach: based on the time-synchronization, OTP also generates password between authentication server and the client. In this algorithm, password is valid for only short period of time. Third approach: the new password is based on a challenge that chosen by authentication server or by client using a mathematical algorithm.

2.2 One-Time Key Generation

One-time key Generation protocol was proposed by Song et. al. [7][10]. One-time key Generation protocol introduced two terminologies local-LMA and home-LMA. This authentication protocol can generate One-time key with Timestamp, Device ID and Key and some special functions as shown in fig. 2.

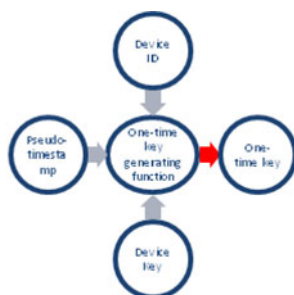


Fig. 2. One-time Key Generation

In their proposed protocol, delivering authentication message from MH to home-LMA will take an extra time because they used pseudo-Timestamp first and they could not transmit Timestamp value with authentication request message for security reason and also they don't have more space for Timestamp in MH-Identifier. So, at the same time MH and home-LMA could not generate One-time Key. To overcome this problem, MH and home-LMA used pseudo-Timestamp that does not match the exact current timestamp. They could get pseudo-Timestamp from simple modulo operation.

3 Proposed Authentication Method for Proxy Mobile IPv6

The One-time Key authentication protocol does not have method to prevent from replay attack and key exposure and it is also time consuming. In order to address the problems, we propose an alternative solution using One-time key Generation with Diameter message to prevent security threats like replay attack and key exposure. To prevent from replay attack and key exposure, we use Diameter message [8] to communicate with backend AAA/Policy server for applications such as network access or IP mobility. Diameter message consists of a Diameter Header that is followed by a number of Diameter attribute value pairs (AVPs). This Diameter Header comprises binary data which is similar to an IP header [9]. AVPs contain AAA information elements and also routing, security and configuration information elements which are relevant to the particular Diameter request or answer message. Each AVP contains some AVP-specific data and an AVP header. Diameter message is also intended to work in both local and roaming AAA situations. We also introduced a terminology LMA/HA configuration of our proposed modified PMIPv6 to reduce the authentication time which is depicted in fig. 3.

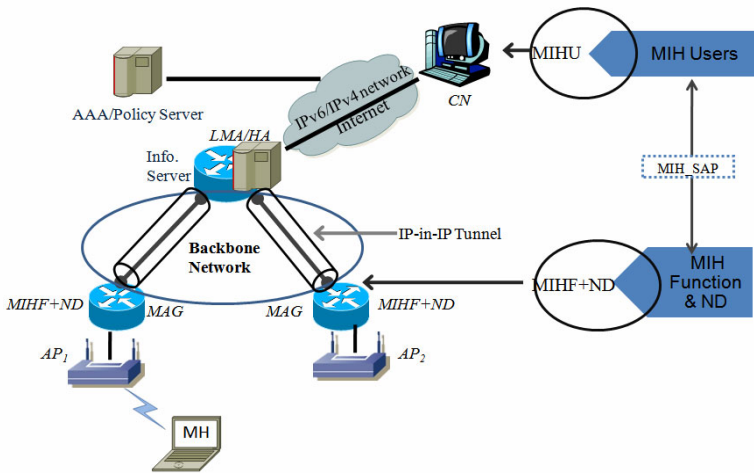


Fig. 3. Proposed Authentication Protocol of PMIPv6-MIH

A MH is identified by its globally unique network access identifier (NAI). When a MH first enters into the PMIPv6 domain, the MH will initiate One-time key generation authentication procedure with the AAA server by sending Mobile Host-Identifier.

3.1 Mobile Host-Identifier (MH-Identifier)

Song et. al. specified the definition of format for MH-Identifier using One-time key [7] [10]. In our proposed protocol, we introduce the similar format for MH-Identifier but using Diameter message. MH-Identifier with Diameter Message format is shown in fig. 4:

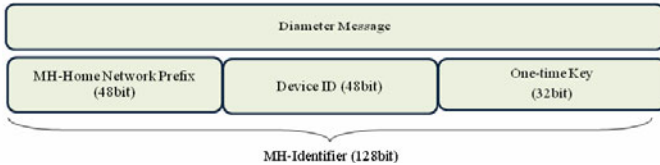


Fig. 4. MH-Identifier with Diameter Message

MH-HNP (48bit)

Mobile Host-Home Network Prefix (MH-HNP) represents the home network prefix of MH. It also introduces a per-MH prefix model in which every MH is assigned a unique address. LMA/HA can find AAA/Policy server with this field. After find the AAA/policy server, LMA/HA send PBU message to AAA/Policy server.

Device ID (48bit)

Typically, Device ID is a MAC address of interface or provides special ID by service provider. This is used to distinguish each MH and for generating properly next field named One-time Key field.

One-time Key (32bit)

One-time Key is the verification field for MH and generated code by the specific random function which is installed both side of MH and AAA/Policy server. Generally, researchers used a time-synchronized type OTP function. There are two approaches for generating this key. One of them is Device ID and the other is current timestamp. The OTP function must have to regenerate One-time Key every few seconds, because sequence of setting up will be done in a few hundred milliseconds. This is one of the main features of this protocol. With this One-time Key, MH can authenticate in simple one-way message from MH to AAA/Policy server and also prevent man-in-the-middle attack because of short time validity of the One-time key.

3.2 Interfacing between MH and MAG

MAG invokes the MH_ATTACH function on MAG when MH attaches to MAG [11]. This function has sub-function that is called MAG_GET_MH_ID. With this sub-function, MAG can get MH-Identifier. During the MH attachment, MAG invokes MIH_Link_up function on MAG.

3.3 Interfacing between MAG and LMA/HA

The authentication mechanism among the MAG, LMA/HA and AAA/policy server must have shared-key security association for communicating securely each other because of some security threats that is described in [5]. As in theoretically, there are lots of MAG than LMAs and number of MAGs are expanding when deployment of PMIPv6 is ongoing. Thus, one PMIPv6 domain has one or several LMAs and one or several MAGs have one LMA. As mentioned earlier, One Time Key generation protocol has two terminologies: (1) It cannot prevent security threats like replay attack and key exposure and (2) It is also time consuming for authentication. In our proposed protocol, we introduced a terminology LMA/HA that means home-LMA and local-LMA are one LMA. On the other hand, LMA is also similar to the HA. LMA/HA are both under same operator's network with MAG that MH is attached and also under the home network of MH. MAG builds up PBU message with the MH-Identifier mobility option for MH and sends it to LMA/HA when MH attaches to MAG. MAG sends RA message to MH with data from PBA if MAG receives positive reply from LMA/HA.

3.4 Interfacing between LMA/HA and AAA/Policy server

When LMA/HA receives PBU from MAG, LMA/HA extracts home network prefix (HNP) from the PBU message and sends Diameter Authentication Request Message to AAA/Policy Server. Using Public Key Infrastructure such as X.509 [12], LMA/HA can authenticate from AAA/Policy Server. The MH-AAA authentication mobility option is used to authenticate the PBU message between the MH and AAA/Policy Server. To verify the PMIPv6 protocol, the mobility message replay protection option is generated and these messages are not replayed by an attacker from some previous message. To compute a session key between MAG and LMA/HA, the key generation nonce request option in the PBU is constructed to request a nonce and that nonce can be stored into the key generation Nonce reply option of PBA. The IPv6 home address request option and the IPv6 assigned home address option are designed to request the Home Address (HoA) of MH.

3.5 Sequences of Authentication Protocol

The sequence of our proposed PMIPv6-MIH authentication protocol is shown in fig. 4:

The sequences of our proposed PMIPv6-MIH authentication signaling are summarized as follows:

Step A:

MH_ATTACH has sub-function such as MAG_GET_MH_ID and with that sub-function, MAG can get MH-Identifier. When MAG receives MIH_Link_up trigger from link layer to IP layer in the MH, the MH sends an authentication request (*AuthReq*) message that contains NAI, identity of MAG and replay protection indicator (RPI) which are used for the AAA/Policy Server to identify the MH and to protect from replay attack. Then a key is computed between MAG and AAA/Policy server called MAG-AAA-KEY when MAG receives the *AuthReq* message. After

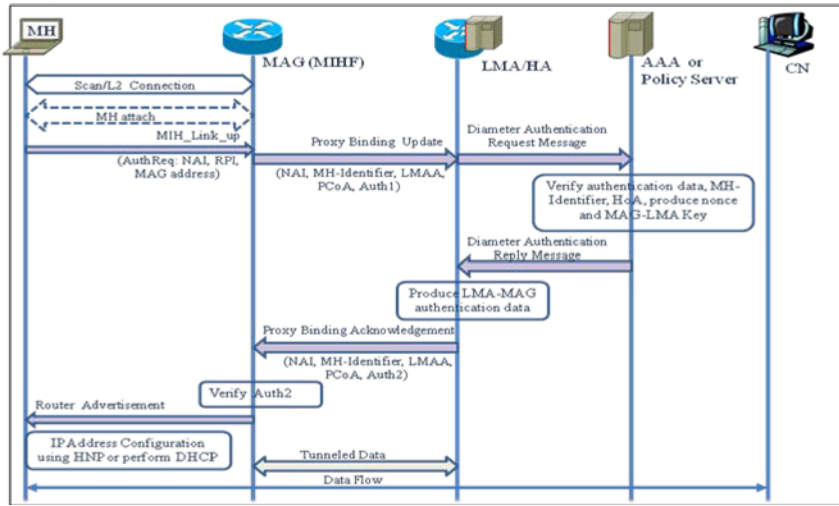


Fig. 5. PMIPv6-MIH Authentication Procedure

authentication, the MAG-AAA key is sent to the MAG. In addition, the MH sends RS message to the MAG to request its home of address (HoA). MAG acquires a PCoA in its PMIPv6 domain. The MAG builds up PBU message with the MH-Identifier mobility option for MH. An authentication data is computed using the MAG-AAA key and is put into the MH-Identifier mobility option of PBU.

$$Auth1 = (\text{Hash-based Message Authentication Code-Secure Hash Algorithm1}) \text{ HMAC-SHA1} (\text{MAG-AAA-KEY}, \text{PCoA} || \text{LMAA} || \text{PBU} || \text{"MAG-AAA-PMIPv6"}) \quad (1)$$

where HMAC-SHA1(K,m) [13] is a keyed hash function computed on message m with key K.

After that, the MAG sends this PBU to the LMA/HA;

Step B:

Upon receiving the PBU message to LMA/HA, it constructs a Diameter authentication Request message which includes many attribute value pairs (AVPs) as follows:

1. PMIPv6-Home-LMA-IPv6-Address
2. MH-Identifier
3. PMIPv6-MAG-Address
4. PMIPv6 Timestamp
5. PMIP Nonce=0
6. MIH Handover Indicator
7. Replay Protection Indicator
8. Access Technology Type

The LMA/HA transmits the Diameter PMIP authentication request message to the AAA server.

Step C:

Upon receiving the Diameter message to AAA/Policy server, it acquires the MH-Identifier and AVP also. It looks up the entire database which stored user identity to identify the requested MH. It also searches the database, if there is Device ID in the subscriber list or not. After that, the AAA generates MH-ONE-TIME key with Device ID and timestamp and verifies whether the timestamp is in the correct range to prevent replay attack. After checking the MH-Identifier data in AAA/Policy server, the AAA can authenticate the MH and verify that the PBU is correct. If all information is valid, then the AAA generates a key generation nonce and computes a session key shared between LMA/HA and MAG.

$$\text{PMIP-MAG-LMA-KEY} = \text{HMAC-SHA1} (\text{MAG-AAA-KEY}, \text{PCOA} \parallel \text{LMAA} \parallel \text{PMIP Nonce}) \quad (2)$$

The AAA/Policy server will construct a Diameter authentication answer message which includes many AVPs as follows:

1. PMIPv6-Home-IPv6-HoA
2. PMIP-MAG-LMA-KEY
3. PMIP-MAG-LMA-KEY Lifetime
4. E (MAG-AAA-KEY, MH-ONE-TIME-KEY, PMIP Nonce)

The AAA/Policy server replies the result to the LMA/HA with the diameter answer message. The key generation nonce is encrypted by the MAG-AAA-KEY.

Step D:

Upon receiving the diameter answer message to LMA/HA, LMA/HA computes the Mobility Message Authentication option of PBA.

$$\text{Auth2} = \text{HMAC-SHA1} (\text{PMIP-MAG-LMA-KEY}, \text{IPv6 HoA} \parallel \text{PBA} \parallel \text{"LMA-MAG-PMIPv6"}) \quad (3)$$

After that, LMA/HA sends this PBA message to MAG.

Step E:

The MAG receives this PBA message. The MAG decrypts nonce and calculates PMIP-MAG-LMA-KEY. The MAG uses this key to verify the correctness of authentication data. If it is valid, the MAG can authenticate the PBA.

The MAG sends Router advertise message with encrypt MH-ONE-TIME-KEY including IPv6 HoA to the MH.

Step F:

The MH decrypts MH-ONE-TIME-KEY and authenticates and also configures IP address using received IPv6 HoA.

Therefore, at the same time our proposed protocol can prevent security threats like replay attack and key exposure, authenticating MH and binding update procedure that are able to reduce authentication latency.

4 Security and Performance Analysis

In PMIPv6 domain, One-time Key authentication protocol (OK-AP) can be used for authenticating MH. However, this authentication protocol cannot prevent from replay attack and key exposure. Therefore, there is a need for an alternative authentication method for PMIPv6-MIH that can protect from replay attack and key exposure.

4.1 Key Exposure

MAG-AAA-KEY is a shared-key association between a MAG and an AAA/Policy server. AAA/Policy server generates a key generation nonce and computes a session key between LMA/HA and MAG called PMIP-LMA-MAG-KEY and also generates MH-ONE-TIME-KEY with Device ID and timestamp to authenticate MH legally. Thus, it is desirable not to leak these keys to the other network entities. The AAA/Policy server construct a Diameter PMIP authentication replay message with encrypts (MAG-AAA-KEY, MH-ONE-TIME-KEY and PMIP nonce) and sends it to the LMA/HA and the MH respectively. The value of key generation nonce encrypted by MAG-AAA-KEY can be decrypted by the MAG and also calculates PMIP-MAG-LMA-KEY, while the other value encrypted by MH-ONE-TIME-KEY is decrypted by the MH. Therefore, MAG-AAA-KEY and MH-ONE-TIME-KEY are not exposed to other entities except the MAG and the MH. With this measure, our proposed protocol is less vulnerable to key exposure.

4.2 Replay Attack

Replay attack involves the passive capture of data and its subsequent retransmission to produce an unauthorized effect. A malicious node keeps an *AuthReq* message to make a false report of normal node and then it can retransmit an old *AuthReq* message to trick the AAA/Policy server for false authentication. In our proposed protocol, this replay attack can be prevented as follows: when MH attaches to MAG, local challenge is created randomly that is a random number for authentication procedure and hence it always changes. Therefore, the malicious node cannot replay the old *AuthReq* message. When even the same local challenge can be selected by the MAG by chance, RPI can prevent the replaying attack.

Table 1. Comparison Analysis of Our Proposed Protocol with OK-AP

	PMIPv6 Authentication Protocol	
	OK-AP	OK-AP with Diameter Message
Auth MH (at home)	YES	YES
Auth MH (at foreign)	YES	YES
Auth LMA/HA	YES	YES
Auth MAG	Possible	YES
One-way Auth	YES	YES
Sniffing-proof	YES	YES

Table 1 shows the comparison results between OK-AP and our proposed protocol OK-AP with Diameter message with some security factors.

5 Conclusion

With the proposed authentication method, not only we are able to reduce authentication latency but also we can prevent security threats like replay attack and key exposure when MH first enters in the PMIPv6 domain. For our future work, we will improve and implement our proposed authentication method on network simulation environment and conduct a more comprehensive security analysis as well as compare with other new authentication mechanism in the PMIPv6 domain.

References

1. Kong, K., Lee, L., Han, Y., Shin, M., You, H.: Mobility Management for All-IP Mobile Networks: Aobile IPv6 vs. Proxy Mobile IPv6. In: Proceedings of the International Conference on Wireless Communications, pp. 36–45 (2008)
2. Lee, H., Han, Y., Min, S.: Network Mobility Support Scheme on PMIPv6 Networks. *International Journal of Computer Networks & Communications (IJCNC)* 2(5) (2010)
3. Taniuchi, K., Ohba, Y., Fajardo, V.: IEEE 802.21: Media Independent Handover: Features, Applicability, and Realization. *Proceedings of IEEE Communications Magazine* 47(1), 112–120 (2009)
4. Narten, T., Nordmark, E., Simpson, W.: Neighbor Discovery for IP Version 6, IPv6 (1998), <http://www.ietf.org/rfc/rfc2461.txt>
5. Vogt, C., Kempf, J.: Security Threats to Network-Based Localized Mobility Management (NETLMM). IETF RFC4832 (2007)
6. Haller, N., Mets, C., Nesser, P., Straw, M.: A One-Time Password System. IETF RFC2289 (1998)
7. Song, J., Han, S.: One-time Key Authentication Protocol for PMIPv6. In: Proceedings of ICCIT 2008, vol. 2, pp. 1150–1153 (2008)
8. Korhonen, J., Bournelle, J., Muhanna, A., Chowdhury, K., Meyer, U.: Diameter Proxy Mobile IPv6: Mobile Access Gateway and Local Mobility Anchor to Diameter Server Interaction. Draft-korhonendime-pmip6-03.txt, Siemens AG, Cisco Systems (2008)
9. Le, F., Patil, B., Perkins, C.E., Faccin, S.: Diameter Mobile IPv6 Application. Draft-le-aaa-diameter-mobileipv6-04 (2004)
10. Song, J., Han, S.: Mobile Node Authentication Protocol for Proxy Mobile. *International Journal of Computer Science and Applications* 6(3), 10–19 (2009)
11. Laganier, J., Narayanan, S., McCann, P.: Interface between a Proxy MIPv6 Mobility Access Gateway and a Mobile Node. IETF netlmm WG Draft (2008)
12. Cooper, M., Dzambasow, Y., Hesse, P., Joseph, S., Nicholas, R.: Internet X.509 Public Key Infrastructure: Certification Path Building. IETF RFC4158 (2005)
13. Krawczyk, H., Bellare, M., Canetti, R.: HMAC: Keyed-Hashing for Message Authentication. RFC 2104 (1997)

A Collaborative Intrusion Detection System against DDoS Attack in Peer to Peer Network

Leila Ranjbar¹ and Siavash Khorsandi²

¹ Islamic Azad University of Qazvin, Daneshgah Street,
Noukhbegan Boulevard, Qazvin, Iran
L.ranjbar@qiau.ac.ir

² Amirkabir University of Technology, 424 Hafez Ave, Tehran, Iran
Khorsandi@aut.ac.ir

Abstract. Peer to peer network is a new form of distribution system and Gnutella network due to its decentralized nature is a special form of peer to peer network. Because of continuous topological changes and the absence of node's dependence on a central unit, security creation is hard to access. In this paper a collaborative intrusion detection system is proposed to detect DDoS attacks by the inspiration of artificial immune system. The function of artificial immune system is distributional, collaborative, robust and adaptive. It increases the precision of attack discovery and decreases false positive rate. Therefore in this paper, after detecting attack with adopting some alternatives, the effect of attack is minimized and the function of the system is optimized. To analyze the function of suggested network in training phase, tcpdump and gtk-gnutella tools are used to create the packets of the Gnutella in a new dataset.

Keywords: Gnutella peer to peer network, artificial immune system, collaborative intrusion detection, DDoS attack.

1 Introduction

Computer networks change and develop very quickly not only in architecture context but also in software context of the network and these changes affect the network traffic. So the examination of the network traffic has always been discussed by researchers. Peer to peer network is very dynamic and it is possible that the topology between peers be different from the point of physical network, also shared files can be replaced according to the topology of peer to peer network. Therefore, traffic in Gnutella hybrid peer to peer network can be examined from different aspects such as, the distribution of packet entrance in time unit, the interval between packet entrance and the distribution of packet size. If the number of these packets exceeds the threshold value, network resources will be saturated and this is because of the nodes (known as servants) in Gnutella hybrid peer to peer network, leave the network or join it anytime. So, they will be exposed to DDoS attacks [13] and such behaviors should be detected and prevented. In order to prevent, detect, encounter and stop attack, security should be recognized.

The first security level is to prevent intrusion and intrusion detection system is the second defensive line [14]. The main strategy to solve security problem in peer to peer network is to use intrusion detection system. By using these strategies, it is possible to detect suspicious ways and potential attacks. As current systems are continuously changing and the strategies to intrude them are also gradually changing, it is essential that intrusion detection system be dynamic over time.

Two noticeable factors in vulnerability of Gnutella hybrid peer to peer network are the flooding sent of message and its decentralized nature [16]. If DDoS attack is managed in Gnutella network, it can be controlled in other peer to peer networks. As DDoS attack contains a large number of distributed machines, the development of defensive nodes would be effective in discovering DDoS attack [13, 17]. Collaborative discovery requires that heterogeneous nodes be adhered and it guarantees high scalability and security against attacks.

Considering the main features of distributed systems and also examining the different mechanisms of human immune system can reveal some similarities between these two seemingly different contexts. Regarding these similarities, we are inspired by human immune system to identify effective intrusion in distributed systems [5, 8]. In the suggested system the combination of artificial immune system algorithms are used. This system follows its operation at several levels with heterogeneous functions of peers.

The rest of this paper is organized as follows. In section (2), we describe the intrusion detection system which is related to this context. In section (3), we briefly introduce human immune system. In section (4), we explore the process of the suggested intrusion detection system and debate surrounding artificial immune algorithms. Section (5), includes a brief analysis of the results and details of our dataset. Finally in section (6), the paper is concluded with a discussion of our proposed intrusion detection system and artificial immune system.

2 Review of Literature

The majority of researches examining attacks just focus on one system but the attacker's purpose is to sabotage several systems. Since the suggested system is based on human immune system, in this section we outline previous studies about the intrusion detection system of Gnutella peer to peer network and also researches that exploit human immune system to secure computer networks.

DD-police protects Gnutella peer to peer network against DoS attack. In this model, peers supervise their neighbors' traffic. Scalability and frequency of sending neighbors' list are two factors that should be mentioned in this model. In peer to peer network with its high dynamic nature, nodes leave and join a lot, and also increase in the frequency of neighbors' list raises the system's overload [12].

In the context of exploiting the features of human immune system for the security of computer networks, Forrest performed the first researches to discriminate between self and nonself in network artificial immune system. Then Hofmeyr designed an artificial immune system called ARTIS. This system is not very efficient because collaboration and information exchange among nodes are not considered and intrusion detection is done separately in each computer.

LISYS is one of the first structures for artificial immune systems that is designed for a simple local network and can learn network traffic and identified anomaly traffic. This system detects seven common network attacks with less than 100 detectors and the length of detector is 49 bits [21].

Cfengine system is the intrusion detection system that uses danger theory and the purpose of it is to automatically configure large number of systems on heterogeneous nodes. Furthermore, as long as a new discordance does not happen, the intrusion detection system is passive. In order to increase scalability, Cfengine intrusion detection system updates the average of system efficiency, the number of each service input and output connection and packet characteristic [4, 10]. Results of Cfengine show that danger signal potentially affects false positive rate and memory detectors improve detection rate.

3 Immune System

The cells and molecules that are responsible for immunity form immune system and their comprehensive and coordinated reply against foreign materials are called immune response. A more comprehensive definition of immune response is the response against microorganisms and macromolecules such as proteins, polysaccharides and small chemical substances that are identified as foreign regardless of physiologic and pathologic outcome of this response [15].

Immune system has two arms for protecting us from foreign agents which constitute innate and adaptive immunity. Innate immunity forms the first line of defense against microbes and it consists of cellular and biochemical defensive mechanisms that exist even before infection and are ready to respond to infections quickly. This mechanism has near equal response against continual infections. Innate immunity mechanisms are unique for the structures that are common among related microbes and they may not distinguish the small differences of nonself.

In contrast to innate immunity, there are other immune responses that are stimulated after exposure to a microorganism. Their defensive power increases after encountering a special microbe. This form of immunity is called adaptable immunity as it evolves in response and also proportionate to infection. Apparent features of adaptable immunity are: enormous response to definite molecules, the ability to remember and stronger response to continual collision to a special kind of microbe [1, 15].

Adaptive immune system is able to identify and respond to a large number of microbe and non microbe substances. In addition, it has a great capacity to distinguish between different microbes and macromolecules even with very close structures. So it is also called exclusive immunity, sometimes in order to put emphasis on the point that strong protective responses are acquired by experience. This system is also called acquired immunity. Main elements of adaptive immunity are lymphocyte and their secretary produces such antibodies. Foreign substances that induce specific immune responses are called antigen.

There are two kinds of adaptive immunity responses, humeral immunity and cellular immunity that operate via different elements of immune system to omit various kinds of microbes.

3.1 Artificial Immune System

De Castro and Timmis [18] define artificial immune systems (AIS) as adaptive systems, inspired by theoretical immunology and supervise immune functions, principles and models, which are applied to problem solving. These systems are developed by inspiration of human immune system not by creating a comprehensive model and they try to capture some or all of the features it provides. In most instances however, only a few principles from immunology are used.

Table 1. Mapping human immune system and Gnutella peer to peer network

Human immune system	Gnutella peer to peer network
Bone marrow and thymus	Intrusion detection system
primary lymphoid organs	Leaf peer
Secondary lymphoid organs	Ultra Peer
Antibody	Detector
Antigen	Intrusion
self	Normal traffic
nonself	Abnormal traffic

3.2 Apparent Features of Adaptive Immunity Responses

All humeral and cellular immunity responses against foreign antigens have some basic features that characterize the lymphocytes which create this response [1, 6]. Generally the features of human immune system that are applied in the proposed system are as follows:

- *Variety*: the total number of lymphocytes antigenic features in a person that are called lymphocyte repertoire is very great. This feature of lymphocyte repertoire is called variety which is the outcome of diversity in the structures of connective areas to the antigen in lymphocyte antigenic receptors. In other words various lymphocyte clones are different from each other in terms of the antigenic receptors structure and consequently antigenic features. So the produced repertoire has a lot of varieties. In the proposed system when Ultra peers detect attack template, they forward it to all connected Ultra Peers and then use genetic algorithm for optimizing attack template. So, we have various attack templates in attack dataset.
- *Memory*: the collision of an immunity system to a foreign antigen increases its ability to respond to the same antigen once again. The responses that are created against the second or next collisions to a kind of antigen are called secondary immunity responses and usually are faster and stronger than the first immunity response against the same kind of antigen. These memory cells have specific features that cause them to operate more effectively than naive lymphocytes that had previous collision to them in responding to antigens and omitting them.

- *Contraction and homeostasis*: after the simulation of antigen, all natural immune responses decrease as time passes. Therefore the immune system returns to repose state and this trend is called constancy or homeostasis. The omission of stimulus causes the death of lymphocytes by means of apoptosis. After detecting attack, Leaf peer goes to suspended mode and then this system returns to repose state.
- *Major histocompatibility cells (MHC)*: Major activities of T lymphocytes consist of defense against intracellular microbes and activation of other cells such as macrophage and B lymphocytes. Therefore the recognition of transplant as self or nonself is a genetic feature. Those genes that are in charge of receiving the transplanted tissues as self or nonself are called histocompatibility between people. All MHC molecules have some specific and common features that are of great importance in presentation of antigen and its recognition by T lymphocytes. In the proposed system negative selection algorithm for training phase running on all Leaf Peers, so it uses MHC properties of human immune system for this purpose.

4 The Proposed Intrusion Detection System

Since the proposed system contains a combination of different algorithms are used to developed purposes, we will investigate this system from three different aspects: intrusion detection system, Gnutella peer to peer network and artificial immune system.

4.1 Intrusion Detection System

As the proposed intrusion detection system is located in all Leaf Peers, system announces the existence of attack or intrusion to other Ultra Peers by means of distributive Ultra Peer warning. Consequently the stated system discovers the network intrusions by cooperation between Leaf Peer and Ultra Peer. Now, we will explore the proposed system from four aspects: detection method, detection on behavior, detection place and detection frequency [22].

A. Detection method

Intrusion detection system distinguishes between behavior-based (also known as anomaly IDS that it compares with normal traffic) and knowledge-based (often known signature-based that it compares with abnormal traffic). To detect intrusion, the algorithms of artificial immune system such as negative selection and clonal selection are used. In fact, new and unknown attacks are detected. Anomaly traffic and normal traffic are distinguished using danger theory. Therefore, the proposed system is formed by the process of combining two methods. In the training phase use anomaly-based intrusion detection and in the test phase utilize signature-based intrusion detection.

B. Detection on behavior

By saturation of network resources in a short time and prediction of attack possibility, the node (Leaf Peer or Ultra Peer) in the suggested intrusion detection system warns its Ultra Peers to confront attacks. Therefore, all surrounding Ultra Peer becomes aware of possible attack. Invaded peers would be suspended since they are not resistant against attack and also they are protected to some extent. This system has an active attitude by detecting and announcing Leaf Peer and Ultra Peer new behaviors.

C. Detection place

Intrusion detection system can be divided into two different groups: network intrusion detection system (NIDS) and host intrusion detection system (HIDS). NIDS is installed on the network’s gateway and examines the traffic of the network from which it passes. Since Ultra Peer in Gnutella hybrid peer to peer network plays the role of gateway and also the role of decider in distinguishing anomaly traffic from normal traffic, the Ultra Peer sends attack strategy to other Ultra Peers after identifying and proving attack.

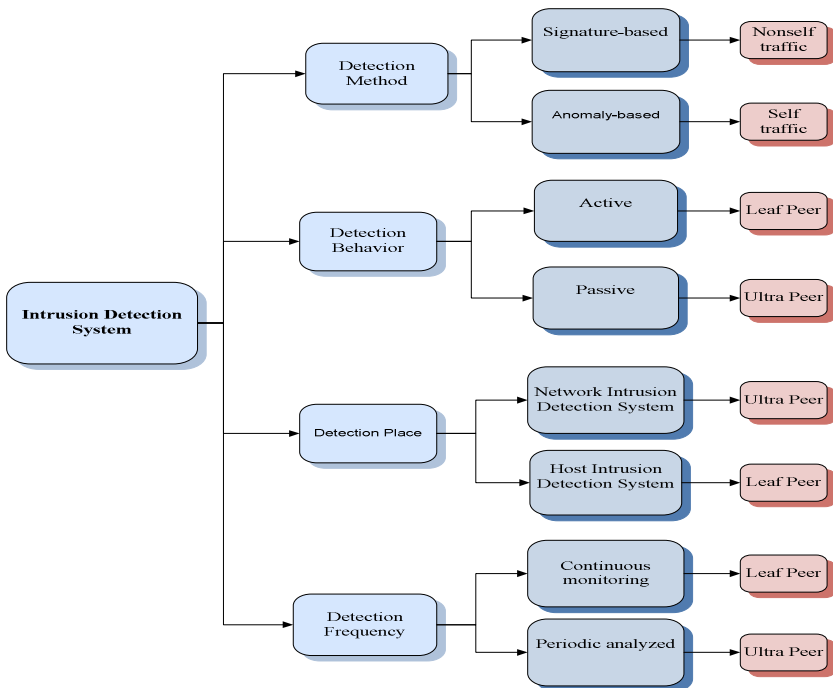


Fig. 1. Taxonomy of intrusion detection system

HIDS performs on different nodes based on collecting network traffic information. These pieces of information are separately analyzed in each node and the results are used to immune the activities of the aforementioned node. Obviously the proposed

intrusion detection system is located on all Leaf Peers. Consequently, this system performs distributively. The results, inform other node of Gnutella hybrid peer to peer network of the existence of attacker node.

D. Detection Frequency

It should be mentioned that Leaf Peers perform intrusion detection continuously but Ultra Peers would be active just by sending the *Stress* message from Leaf Peers.

4.2 Peer to Peer Network

Peer to peer networks are divided into two groups, structured and unstructured groups but new generation of peer to peer networks combine these groups and inherit their advantages. In this paper, we focus on Gnutella hybrid peer to peer network. This system works based on Gnutella traffic. We have improved network function to detect intrusion by adding some messages to Gnutella's main messages. Each message is mentioned and explained below.

Gnutella peer activity is divided into two groups, signaling and data transmitting. Experience shows that signaling profusion is the main threat of Gnutella network against invasion and attack such as DDoS. Apparently signaling in Gnutella network is created with the exchange of messages to explore topology and discover resources [12]. Now, we discuss new messages to add Gnutellasm simulation:

Stress message: if 70 percent of the used Leaf Peer's bandwidth saturate, it will inform its Ultra Peers of the possibility of attack by stress message.

Stress Reply message: Ultra Peer becomes aware of possible attack by receiving Stress message and it receives attack template from Leaf Peer by sending Stress Reply. If Leaf Peer does not receive Stress Reply, it will not send the template and may be disconnected.

Template message: By Template message, the template of possible attack is sent to Ultra Peer.

Maintemplate message: Ultra Peer compares attack template with the previously identified attacks' templates. At most 30% of templates conform (to r-contiguous bits matching [22]), definite attack occurrence will be announced to other Ultra Peers by Maintemplate message.

4.2.1 The Proposed System Related Function

This system uses different functions to detect intrusion or DDoS attack. Each peer may have more than one function. To create alarm in the proposed system, a process should be followed that requires several functions mentioned below:

1- *The function of creating Template:* Leaf Peer records the template of messages that it receives in a short time and if volume of messages is more than threshold in this span, a template is formed by the source IP address, the destination IP address (local) and time interval between Gnutella packets and then it is sent as the template of possible attack; otherwise the produced template is removed.

2- *The function of sending and receiving the possible attack occurrence template:* after possible attack template is formed by Leaf Peer, other peers should be informed of the possible occurrence of this attack. If Ultra Peer returns Stress Reply message, Leaf Peer will inform that of the possible attack occurrence by sending Stress message. The possible attack template is sent to Ultra Peer by Template message.

3- *The function of identifying attack based on received template from Leaf Peer:* after receiving the possible attack template by Template message, Ultra Peer starts the activity of conforming received template to the template of available attacks in dataset. 30 percent conformity shows that an attack has happened.

4- *The function of sending attack template to other Ultra Peers:* after discovering attack, Ultra Peer sends attack template to other Ultra Peers. So they would be informed of attack occurrence and increase their detectors.

5- *The function of Classification:* to be able to distinguish between anomaly traffic and normal traffic, an attitude should be chosen that considers the peer sent traffic as attack traffic or anomaly traffic by receiving numerous Gnutella messages in definite time intervals and also by saturating bandwidth,. In two steps, we used classification function. In first step Leaf Peers distinguish between normal traffic and possible abnormal traffic. This process is named discrimination self/nonself and in second step, Ultra Peers distinguish between possible abnormal traffic and abnormal traffic, this process is done by danger theory [10].

6- *The function of Rate limit:* if the number of sent messages is more than bandwidth occupation threshold and possibility of attack occurrence is announced, sending and receiving message to the Ultra Peer can be prevented and the rate of sent messages can be reduced by adopting some measures. In fact invaded peers would be suspended since they are not resistant against attack and they are protected to some extent, in a way that they just accept high priority packets that are sent by surrounding Ultra Peers.

7- *The function of producing new generation of detector (Genetic algorithm):* Therefore, the templates in which the most conformity has happened are most likely to happen in future and such templates are used in the selection phase of genetic algorithm. In fact ranking method is used, in a way that detectors are first ranked based on number of conformities and then template selection will be done according to rank based fitness.

On this condition category conformity is possible and competitive method is used among the best attack templates for selection. This method works in a way that a small subcategory of attack templates is randomly chosen and then competes together. Finally in this competition, one of them is chosen based on affinity level. After selecting best templates (with more conformity) by crossover operator and with the purpose of producing better templates, new templates were created. After the function of attack templates crossover, mutation includes the change of zero to one. On the other hand, the function is applied in a lymphocyte repertoire to protect the different forms of the distinction of attack templates.

4.3 Artificial Immune Algorithm

Since human immune system performs actively and distributively, artificial immune system algorithms are extremely used in proposed system to develop our purpose. Here major features of human immune system are inspected to detect intrusion and how it reacts against intrusions [11]. Then its application in distributed peer to peer network to confront DDoS attack will be mentioned. In the suggested system negative selection algorithm is used in training phase and its function is as follows:

Network normal traffic which contains Gnutella network packets is captured by tcpdump monitoring tools [3] and gtk-gnutella file sharing software [2]. Then it is considered as a self dataset. After that some detectors (immature detectors) are produced by random Gaussian function and by comparing these two datasets, any detector that does not correspond to network's normal traffic will be added to the detectors' list as nonself detector (mature detectors). At this stage, the number of detectors is investigated. If this number increases, the accuracy of detection goes up and computational overload increases too [7].

Algorithm 1. Using Negative Selection in training phase

```
01:  $G_{nd}$ : Gnutella normal dataset
02:  $G_{ad}$ : Gnutella abnormal dataset (detector dataset)
03:  $d$ : detector
04:  $D_{th}$ : Threshold of detector
05: while number of  $d$  less than  $D_{th}$ 
06:    $d \leftarrow$  create immature detector with uniform Gaussian random function
07:   if  $G_{nd}$  contains  $d$  then
08:     drop  $d$ 
09:   else
10:      $d$  insert into  $G_{ad}$ 
11:   end if
12: end while
```

After receiving each Gnutella packet, the source IP address, the local destination IP address and also the average time interval between two consecutive in sent packets will be added to template. Then the size of bandwidth occupation will be examined. If it does not reach the default threshold, the template will be faded out of existence and a new template will be made.

Otherwise, the possibility of attack occurrence will be announced to connect Ultra Peers and then Leaf Peer after making sure of the existence of each Ultra Peer, sends the template of possible attack to them. At this stage, Leaf Peer announces the possibility of attack occurrence and distinguishes between abnormal traffic and normal traffic. Leaf Peer will be suspended in a definite time span to prevent the reception of any packet or message. When this time span ends, Leaf Peer will return to its initial state.

Ultra Peer announces its existence to Leaf Peer by receiving the possibility of attack occurrence and after receiving the template of possible attack compares that to its nonself

dataset. If the template conforms to each detector, Ultra Peer broadcasts it to other Ultra Peers as a detector. Then Ultra Peer creates conformed detectors once again, increases their affinity and if detectors aren't conformed, Ultra Peer will make them older. In either way Ultra Peer examines detectors' affinity in order to change its main structure.

Algorithm 2. Leaf Peer Function in test phase

```

01:   $G_p$ : Gnutella Packet
02:   $BW_d$ : percentage of Leaf Peer Bandwidth depletion
03:   $BW_{th}$ : Threshold of Leaf Peer Bandwidth depletion
04:  While peer is in active mode
05:     $T \leftarrow$  receive features of new  $G_p$ 
06:    if  $BW_d \geq BW_{th}$  then
07:      forwards msg-stress along connected Ultra Peers
08:    else
09:      Drop  $T$ 
10:    end if
11:    if received msg-sressreply then
12:      forwards  $T$  to certain Ultra Peers
13:      stand in suspend mode for time span
14:    end if
15:  end while

```

Algorithm 3. Ultra Peer Function in test phase

```

01:   $T_a$ : Template of attack
02:   $T_c$ : number of conformity with  $T_a$ 
03:   $T_{ttl}$ : time to live for every detector
04:  while Ultra Peer is in active mode
05:     $T \leftarrow$  receive  $G_p$ 
06:    if  $G_p$ .Type is msg_stress then
07:      forwards msg_stressreply along Leaf Peer
08:    end if
09:     $T_a \leftarrow$  received msg_template
10:    if  $G_{ad}$  contains  $T_a$  then
11:      increment  $T_c$ 
12:      set  $T_{ttl}$  to zero
13:      update  $G_{ad}$  with  $T_a$ 
14:      forward  $T_a$  along every Ultra Peers in network
15:      Run GA .Algorithm on  $G_{ad}$ 
16:    end if
17:  end while

```

According to the number of conformities, detectors' situation changes from mature stage to active stage and from active stage to memory stage. On next step each detector's beneficial life time along with its kind is inspected. As each kind of detector has a definite life time, those detectors whose life time is ended are deleted from detectors dataset. Genetic algorithm is used to improve detectors in the proposed system. Genetic algorithm also causes variety in nonself templates at active stage, in a way that based on clonal selection algorithm, those cells that identify detector grow and those cells that are not able to identify detector die. As Leaf Peer and Ultra Peer operate in a collaborative and parallel manner and available network peers are fully distributed, Leaf Peer's and Ultra Peer's function are separately inspected.

5 Simulation Study

We implemented a discrete event simulator of a Gnutella peer to peer file sharing. Gnutellasim is suitable for Gnutella network and is installed on PDNS and ns2.27. In order to evaluate the suggested system, gtk-gnutella-0.96.8-2 file sharing client [2] and tcpdump-4.1.1 monitoring software [3] is used to generate and record Gnutella traffic.

5.1 Simulation Data Preliminaries

One challenge in intrusion detection is to find good data sets for experimentation and testing. Our objective was to control the data set, so we chose to collect data from an internal restricted Gnutella peer to peer network. In this environment, we can understand all of the connections, and we can limit DDoS attacks. We install firewall of ISA server at the entrance of our network. Then external connections must pass through a firewall.

The Dataset used in this paper is related to Gnutella peer to peer network traffic. The proposed scenario includes 23 peers that are divided into 5 Ultra Peers and 18 Leaf Peers.

5.2 Simulation Results Analysis

In this intrusion detection system, self is defined to be the set of normal pair wise TCP/IP connections between Leaf Peer and Ultra Peer (for Gnutella 0.6) and nonself is the set of connections, an enormous number Gnutella packets are transmitted which are not normally observed on the network. So, the efficiency of proposed system is analyzed based on the following criteria:

1- Negative selection time: Some immature detectors are produced by random Gaussian function and this dataset is compared with Gnutella normal dataset. If any detector does not match normal traffic template, it will be added to the mature detectors' list. Output of training file is a mature detectors' dataset. Figure 2 shows the time of negative selection in proportion to the number of detectors. When the number of mature detectors increases, negative selection time will increase too and detection precision will be optimized. Because of using genetic algorithm, the time of negative selection is more beneficial than LISYS algorithm.

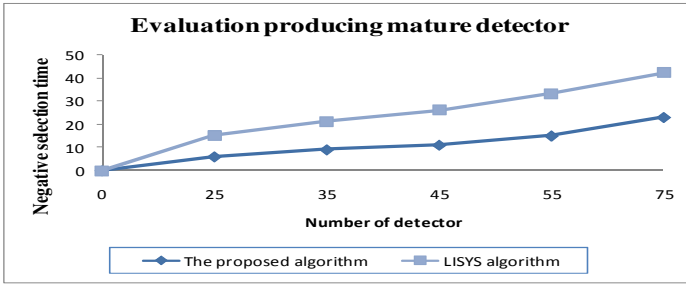


Fig. 2. The production time of mature detector

2- *Detection Precision*: In order to increase detection precision, false positive should be reduced. These parameters include:

- Number of detectors
- r parameters of bit matching algorithm
- activation threshold parameters

We also look for parameters that appear to be most important for minimizing false positives, as well as how percentage of detecting intrusions maximizes. The percentage of attack detection will be measured by the proportion of discovered attack occurrences to all attack occurrences. R_{dt} denotes the corresponding false positives rate. T_d is the number of attacks that were discovered and T_a is the total number of attacks.

$$R_{dt} = \frac{T_d}{T_a} \times 100 \tag{1}$$

In fact «false positive is the sending of alarm message by intrusion detection system at the time that attack has not happened». T_p is the total number of false positive alarms and T_a is the total number of attacks.

$$R_{fp} = \frac{T_p}{T_a} \times 100 \tag{2}$$

The proposed system is adopted to describe the tradeoff between the detection rate and false positive rate. Therefore, we evaluate the best attitudes that are coherent to these factors for yielding optimum resolves.

A. number of detectors

To study the effect of mature detectors on the percentage of attack discovery and false positive, the parameter of activation threshold is considered 6, crossover operator 0.4 and mutation operator 0.005. These two factors are evaluated by the change in the number of detectors and different conformity bits. Through increase in the number of detectors, the percentage of attack discovery goes up on the one side and the false positive increases on the other side. In a way that in all forms of conformity bit, 75

detectors show the most efficient response for detecting attack. But due to computation overload, the number of detectors is commonly not very high. In LISYS algorithm, the number of detectors is 100. Figure 3 proves this.

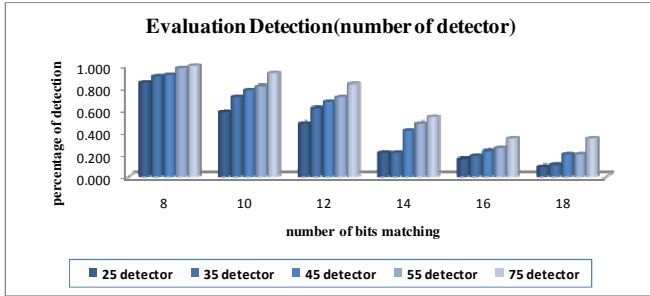


Fig. 3. Evaluation detection with different number of detector

B. r parameters of bit matching algorithm

Some detectors in this system usually implement as strings, whose function is to classify new strings as normal or abnormal by matching them in some forms. The perfect matching is rare in the immune system. So, we use a partial matching rule is known as r-contiguous bits matching. Under this rule, if two strings match in at least r contiguous locations, they are identified as attack template.

Our observations in figure 4 show that immune system is the best approach for detecting intrusion. In particular, the r-contiguous bits matching rule is proposed in LISYS and we use it for our system. To study the effect of mature detectors on the percentage of attack discovery and false positive, the parameter of activation threshold is considered 6, crossover operator 0.6 and mutation operator 0.005. These two factors are evaluated by the change in the number of detectors and different conformity bits. The number of strings in which a detector matches, increases exponentially as the value of r decreases. For example, 8 conformity bits is the best resolve for attack detection rate but is the worst result for false positive rate. After checking these factors, we select 12 conformity bits and LISYS algorithm selects a number, too.

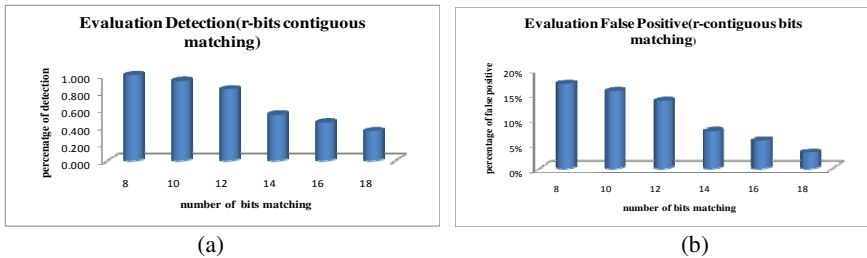


Fig. 4. Evaluation Detection and Evaluation false positive

C. Activation threshold parameter

Activation threshold shows detector’s condition in mature, active and memory state. Activation thresholds are a mechanism designed to reduce false positives. To test our expectations, we studied the effect of changing the activation threshold on the number of false positives. These experiments were run with different r . The proper amount of activation threshold is evaluated with 75 detectors, crossover operator 0.6 and mutation operator 0.005.

The sooner the detector goes to activation stage, the more the generation production will be and the better discovery will occur. Also this parameter decreases the false positive. Figure 5 illustrates how the number of false positives lessens as the activation threshold increases.

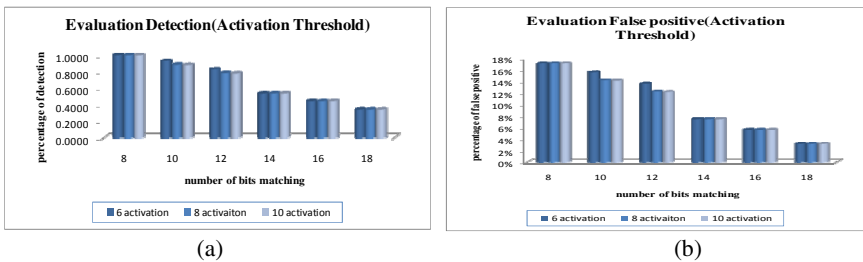


Fig. 5. Evaluation Detection and Evaluation false positive

6 Conclusion

Since creating security in distributed networks is complicated, in order to obtain the maximum of security and possible attacks discovery, it is required to use the advantages of various methods of intrusion detection. The proposed systems which are used for the two approaches of intrusion detection are anomaly-based and signature-based. Each time an attack is identified by Genetic algorithm, a new generation is created and then it will be added to detectors dataset. In fact the template of new attacks is discovered and this approach increases the ability and power of the system. When false positive decreases, attack detection precision increases. The proposed system in this paper with consideration the adopting some alternative, minimizes the influence of attack after its identification and makes system function efficient even more than an accepted level. In addition, the proposed system inspects nodes cooperation and how to use the algorithms of artificial immune system. The results of simulation show that the method which is used for this purpose, not only has adaptability, scalability, flexibility and variety but also has high accuracy and correctness.

References

[1] Artificial immune system(AIS), <http://www.artificial-immune-system.org>
 [2] gtk-gnutella, <http://www.gtk-gnutella.com>

- [3] tcpdump, <http://www.tcpdump.org>
- [4] Aickelin, U., Bentley, P., Cayzer, S., Kim, J., McLeod, J.: Danger Theory: The Link between Artificial Immune Systems and Intrusion Detection Systems. In: Proceedings 2nd International Conference on Artificial Immune Systems, pp. 147–155 (2003)
- [5] Aickelin, U., Greensmith, J., Twycross, J.: Immune system approaches to intrusion detection – A review. In: Nicosia, G., Cutello, V., Bentley, P.J., Timmis, J. (eds.) ICARIS 2004. LNCS, vol. 3239, pp. 316–329. Springer, Heidelberg (2004)
- [6] Okine, A., Dasgupta, D., Nii: Immunity-based systems: A survey. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, pp. 369–374 (1997)
- [7] Bentley, P.J., Kim, J.: Evaluating negative selection in an artificial immune system for network intrusion detection. In: Proceedings of GECCO 2001, pp. 1330–1337 (2001)
- [8] Bentley, P.J., Kim, J.: Towards an artificial immune system for network intrusion detection: An investigation of dynamic clonal selection. In: Congress on Evolutionary Computation (CEC 2001), Seoul, Korea, pp. 1244–1252 (2001)
- [9] Twycross, J., Aickelin, U.: Towards a Conceptual Framework for Innate Immunity. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, pp. 112–125. Springer, Heidelberg (2005)
- [10] Cayzer, S., Aickelin, U.: Danger theory and its applications to AIS. In: Proceeding of the Second International Conference on Artificial Immune Systems (ICARIS 2002), pp. 141–148 (2002)
- [11] Chang, R.: Defending Against Flooding-Based Distributed Denial-of-Service Attacks. IEEE Communications Magazine, 42–51 (2001)
- [12] Athanasopoulos, E., Anagnostakis, K.G., Markatos, E.P.: Misusing unstructured P2P systems to perform doS attacks: The network that never forgets. In: Zhou, J., Yung, M., Bao, F. (eds.) ACNS 2006. LNCS, vol. 3989, pp. 130–145. Springer, Heidelberg (2006)
- [13] Oikonomou, G., Reiher, P., Robinson, M., Mirkovic, J.: A framework for collaborative DDOS defense. In: Proceedings of the 2006 annual computer security applications conference, pp. 33–42 (2006)
- [14] Guan, J., Liu, D.X., Cui, B.G.: An induction learning approach for building intrusion detection models using genetic algorithms. In: Proceedings of Fifth World Congress on Intelligent Control and Automation WCICA, vol. 5, pp. 4339–4342 (2004)
- [15] Kephart, J.: A biologically inspired immune system for computers. In: Proceedings of the Fourth International Workshop on Synthesis and Simulation of Living Systems, Artificial Life IV, pp. 130–139 (1994)
- [16] Xiao, L., Liu, Y., Ni, L.M.: Improving Unstructured Peer-to-Peer Systems by Adaptive Connection Establishment. IEEE Transactions on Computers (2005)
- [17] Mirkovic, J., Prier, G., Reiher, P.: Alliance formation for DDoS defense. In: Proceedings of the New Security Paradigms Workshop, ACM SIGSAC, Ascona, Switzerland, pp. 11–18 (2003)
- [18] Stepney, S., Smith, R.E., Timmis, J.I., Tyrrell, A.M.: Towards a conceptual framework for artificial immune systems. In: Nicosia, G., Cutello, V., Bentley, P.J., Timmis, J. (eds.) ICARIS 2004. LNCS, vol. 3239, pp. 53–64. Springer, Heidelberg (2004)
- [19] Singh, S.: Anomaly detection using negative selection based on the r-contiguous matching rule. In: Proceedings of the 1st International Conference on Artificial Immune Systems (ICARIS 2002), pp. 99–106 (2002)
- [20] Aickelin, U., Bentley, P.J., Cayzer, S., Kim, J., McLeod, J.: Danger theory: The link between AIS and IDS? In: Timmis, J., Bentley, P.J., Hart, E. (eds.) ICARIS 2003. LNCS, vol. 2787, pp. 147–155. Springer, Heidelberg (2003)
- [21] Balthrop, J., Forrest, S., Glickman, M.: Revisiting lysis: Parameters and normal behavior. In: CEC 2002: Proceedings of the Congress on Evolutionary Computing (2002)

Impact of Safety Beacons on the Performance of Vehicular Ad Hoc Networks

Bilal Munir Mughal, Asif Ali Wagan, and Halabi Hasbullah

Department of Computer and Information Sciences,
Universiti Teknologi PETRONAS,
Bandar Seri Iskandar, 31750 Tronoh, Perak, Malaysia
bilalmunirmughal@gmail.com,
asifwaggan@gmail.com,
halabi@petronas.com.my

Abstract. Saving human lives on road has become the prime objective of Vehicular Adhoc Network (VANET). VANET-equipped vehicles broadcast periodic safety beacons (single-hop) and event-driven messages (multi-hop) to keep the neighboring vehicles aware of the situation at all times. Well known broadcast communication problems i.e. hidden/exposed nodes, collisions and inherent challenges in VANET e.g. dynamic environment, limited bandwidth, are likely to hinder the exchange of potentially lifesaving information. It is also known that much of the vehicle-to-vehicle broadcast communication will comprise of single-hop periodic safety beacons thus it becomes important to analyze their impact on VANET performance. In this study, extensive simulations were conducted out to appraise the performance of single-hop periodic safety beacons in the context of defined parameters i.e. communication range (CR), beacon generation interval (BGI) and safety beacon size (SBS). The Quality-of-service (QoS) metrics used for the evaluation are packet delivery ratio (PDR), per-node throughput, and end-to-end delay.

Keywords: VANET; safety beacons; performance evaluation.

1 Introduction

Vehicular ad hoc network (VANET) is essentially a part of Intelligent Transportation System (ITS). By definition VANET is a form of Mobile Ad hoc Network (MANET), in which vehicles form a decentralized network by communicating via On-board units (OBUs). One main distinction between MANET and VANET is that in MANETs nodes move arbitrarily and in VANETs nodes primarily follow a predefined path such as roads making their movement more conventional. Generally two types of communication takes place in VANET i.e. vehicles communicate with roadside infrastructure (V2R) and with nearby vehicles (V2V), collectively described as vehicle to all (V2X) communication. VANET applications can be divided into two major categories i.e. safety and non-safety applications. Applications that are critical to human life safety are placed under safety application category and the rest of the applications lay in non-safety category.

Currently in VANET domain much of the researches have been done to help define standards, i.e. frequency allocation, PHY & Link layer standards, security issues and new applications [1]. In USA, Federal Communications Commission (FCC) has allocated Dedicated Short Range Communication (DSRC) spectrum at 5.9 GHz, which is structured into seven of 10 MHz channels. Channel 178 (5.885-5.895GHz) is the control channel (CCH) which is to be primarily used for safety communication including both single-hop periodic beacons and event-driven multi-hop messages. At PHY level, the philosophy of IEEE 802.11p design is to make minimum necessary changes to IEEE PHY so that WAVE devices can communicate effectively among fast moving vehicles in the roadway environment [2]. Some of the application characteristics have been defined in Vehicle Safety Communications project report [3]. However safety application design is still an ongoing research area.

Providing efficient safety messaging scheme is a challenging task due to some specific characteristics of VANET i.e. high mobility, limited channel bandwidth, very short communication duration, and highly dynamic topology. Furthermore the broadcast nature of communication in VANET, may lead to saturated/congested channel, which was identified as a major concern for efficient safety communication by [4] and [5]. However it is possible to reduce side effects by taking appropriate remedial actions, for example according to [6] transmission powers and transmission rate are suitable methods for periodic message congestion control.

It is understood that single-hop periodic safety beacons (SBs) predominantly occupy the control channel communication and may easily consume entire available bandwidth. Up till now, most of the previous work is focused on multi-hop communication. Thus for developing efficient safety messaging schemes it is essential that effects of single-hop SBs on overall VANET performance be known beforehand. Furthermore it is also necessary to evaluate the parameters involved in controlling the behavior of periodic safety beacons i.e. beacon generation interval (BGI), safety beacon size (SBS), and communication range (CR)/transmission power. In this study, extensive set of simulations are carried out to better understand the impact of single-hop periodic safety beaconing on VANET performance and also to gain insight into tunable parameters that control periodic beaconing. Packet delivery ratio (PDR), per-node throughput and end-to-delay (e2e delay) are quality of service (QoS) metrics used for the evaluation. Successful and timely delivery of SBs is essential for saving lives in potentially dangerous situations on the road. The simulation results presented in this study can be potentially helpful for VANET application and standard development.

The rest of the paper is organized as follows. Section-2: related work, Section-3: Research Methodology, Section-4: Simulation Setup, Section-5: Results and Analysis, finally this paper is concluded in section-6 followed by references.

2 Related Work

A broad review regarding VANET communication challenges is given in [7]. VANET primarily uses broadcasting as the basic communication mechanism. Safety beacons are broadcasted in one hop range while event-driven messages can be disseminated over multiple hops. Multi-hop broadcasting, i.e. flooding has been

extensively studied in the literature [8-11]. However one-hop broadcasting has been of lesser focus in VANET studies.

Broadcast being the primary communication mode, the channel congestion is an anticipated problem that many previous studies have addressed. Lars Wischhof and Hermann Rohling provide a Utility-Based Packet Forwarding and Congestion Control Scheme (UBPFCC) [12] that works on top of IEEE 802.11 MAC protocol and is focused only on non-safety applications. This approach needs the road to be segmented for calculating the message utility metric, thus it cannot be readily applied to VANET. Congestion detection along with safety beacon rate control algorithms is presented in [13].

Many researchers rely on the power control techniques to increase packet reception and mitigate channel congestion. Authors of [14] present a power control scheme based on estimation of surrounding traffic density concerning a particular node. However, the main focus is to maintain connectivity using dynamic transmission range assignment. Marc Torrent-Moreno et al [15-17], present another power control method called Distributed Fair transmit Power Adjustment for Vehicular ad hoc networks (D-FPAV) for Vehicular environment. In this method vehicles have to adjust their transmission power using in such a way that bandwidth utilized by periodic messaging does not exceed a predefined threshold called MBL (maximum beaconing load). The idea behind defining MBL is to reserve a chunk of bandwidth for event-driven message so that communication of event-driven messages is not hindered by channel saturation. In a fully converged D-FPAV based VANET all nodes should have a minimum common power level that may not be a suitable choice in diverse traffic densities and high mobility. To reduce communication overhead generated by D-FPAV, Jens Mittag et al. introduce Distributed vehicle Density Estimation (DVDE) and Segment-based Power Adjustment for Vehicular environments (SPAV) strategies [18]. Simulation results of DVDE/SPAV also confirm less control overhead as compared to D-FPAV. It is also shown that in order to guarantee a network-wide beaconing threshold, precise information is necessary up to three times a node's maximum communication range, regardless of the propagation model or of traffic density distribution. Aggregating such information with high accuracy is very difficult to achieve in a challenging environment i.e. VANET.

Authors in [19-20] proposed a power adaptive algorithm based on an analytical model to maximize 1-hop broadcast area using CSMA. However as mentioned before same transmission power for all nodes is not suitable to accommodate varying density and dynamic environment. Another power control technique is introduced in [21] that is based upon a Delay-Bounded Dynamic Interactive Power Control module that shows prompt 1-hop neighbor connectivity but makes use of eight directional antennas thus not readily suitable for VANET environment.

Most of the studies mentioned above have either partially explored the periodic safety beaconing effects on VANET or simply proposed performance enhancement schemes based on general assumptions regarding broadcast communication behavior. Thus these studies do not fulfill the requirements for in-depth analysis of single-hop periodic safety beaconing.

Perhaps the most closely related work to this study is [22-24], in which the authors performed simulation studies for exploring some predefined VANET message dissemination characteristics. Priority access is the main focus of [22] and evaluation

parameter used is one hop broadcast message reception rate. As the focus is priority access evaluation, simulation are carried out with limited configurations i.e. communication range of 100m, 200m and packet size of 200B, 500B only. Different data rates and somewhat similar communication range, packet size for simulation settings are chosen by the authors of [23], which is also one the earliest works in this area. Furthermore evaluation parameters used are probability of reception failure and channel busy time. Simulations performed in both of these studies basically use earlier version of NS-2 with several shortcomings in 802.11 MAC and PHY layers e.g. the inability to handle collisions, path loss calculations and interferences. A detailed analysis on the shortcomings of 802.11 in previous versions of NS-2 and comparison of 802.11a/802.11p can be found in [24], [25] respectively.

Yousefi et al. use different adjustable network parameters in [26] i.e. power (communication range), packet size and packet dissemination interval, which is similar to current study. However their choice of values for these parameters is an important factor to look into. Such as simulating packet size of 100 and 200 byte only is not practical, according to [27] actual message size will be rather large i.e. between 280 to 800 bytes due the incurred security overhead. Furthermore a communication range of up to 300m is a reasonable choice in jammed traffic scenarios but does not cover various traffic situations where a wider range is required e.g. sparse traffic conditions. Similarly 100ms and 200ms packet dissemination intervals do not provide significant insight into the overall behavior of the parameter which we find to be very important factor for enhancing the performance of VANET in terms of packet reception (discussed later).

Limitations mentioned above provide the motivation for this research. In this study, results from extensive set of simulations are presented to broadly analyze the impact of adjustable parameters that notably impact the performance of VANETs. Moreover simulations in this study are performed using enhanced 802.11 NS-2 [24] module that provides more realistic VANET MAC and PHY layer thus giving more accurate results. Additionally, the obtained threshold values of adjustable parameters shall be used for our previously proposed congestion control scheme [28].

3 Research Methodology

Traditionally in wireless communication and specifically under broadcast environment Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is prone to higher collision rate in the presence of well known hidden and/or exposed node affect. This could lead to a number of adverse effects on the network i.e. packet loss, low throughput and higher delays. In the absence of RTS/CTS (Request to send/Clear to send) situation can be more adverse for VANETs. Thus there is a strong need to explore potential parameters that can hinder safety communication in VANET.

Single-hop periodic safety beacons (SBs) will dominate channel load in VANETs. It is essential to evaluate their behavior and impact on overall network performance. It is also important that the evaluation is done on the basis of parameters that govern SBs. Furthermore reliable evaluation should be carried out in the light of some

relevant quality of service (QoS) metrics. A brief introduction to the tunable parameters and QoS metrics used for this study are given below.

3.1 Tunable Parameters

Parameters explored in this study are beacon generation interval, transmission power and packet size that can influence overall performance of the network. The brief descriptions of these parameters are given next.

Transmission Power/Communication Range: It is the most commonly used parameter in the literature for performance optimization. Transmission power can be increased or decreased to reduce collisions. Decreasing the transmission power essentially means reducing the number of nodes competing for the same channel as in VANET. In typical road situations node distribution is unpredictable and is mostly heterogeneous in nature thus having a common power level among the nodes at broader level is not practical. Consequently it is more useful for each node to adjust its power according to immediate neighborhood situation. For example setting minimum or maximum common power level for a road segment of certain length having higher node density at the centre and lower node density at the edges, may result in isolation of farther nodes or higher collisions at the centre. For VANET, maximum transmission power corresponding to a communication range of 1000m is desirable while lower bounds vary according to underlying application requirement in different road scenarios.

Beacon Generation Interval (BGI): The rate at which a node generates messages per unit of time is known as Beacon Generation Interval (BGI). BGI remains a relatively less explored parameter mainly because of the considerations that longer BGI may cause higher communication delays which can lead to ineffectiveness of safety applications. It is generally assumed that DSRC supported vehicles will exchange Safety Beacons every 100msec. However a realistic BGI should account for human reaction time, vehicle speed/acceleration, positioning update frequency of GPS equipment and propagation delay.

According to [29], mean human reaction time for close encounters is 700ms or higher, anything beyond this point may have no practical use as the driver is able to react faster than the VANET itself.

VANETs are expected to support vehicle speed of up to 120mph or approx. 193km/h. At maximum intended speed a vehicle can travel 53.61m in one second which seems a considerable change of position. However when sending the periodic SB at every 100msec the actual distance covered by the sender in the mean time is only 5.36m which is less significant considering the speed it is traveling at. Similarly decreasing speeds means that there will be even smaller variations in senders traveled distance between two consecutive Safety Beacons. Consequently at lower speed it becomes feasible to increase the time delay between two consecutive SBs. Furthermore, with assistance from built-in maps and sender information (e.g. sending time of the beacon, speed/acceleration and intended destination) available in SB itself it is possible for receiving nodes to predict the current position of the sending node even between the reception of two concurrent SBs.

VANET-ready vehicles need to get their positioning information from the low cost GPS equipment, positioning update frequency in such systems is usually 200ms or slower. However GPS devices with much faster update frequencies are available at higher costs.

Based on this fact we can safely assume that an upper bound of 500ms and less can still provide a viable assistance to the driver.

Beacon Payload Size: Amount of actual information in a beacon excluding the headers. Size of the beacons to be exchanged in any network is of great importance. In VANET beacons may carry various types of information including velocity, position and hazard information. As a general concept more information carried by a beacon means a well informed neighborhood with better safety. However, increasing beacon size contributes towards channel saturation which is certainly not a desired feature in any network especially in CSAM/CA based VANETs.

3.2 Evaluation Metrics

Since broadcast is the basic method used for periodic beaconing, average or mean values of results obtained from multiple nodes are better suited for accuracy. Unless otherwise specified, all resulting values are based upon averages obtained from selective nodes.

Per-node Throughput: Number of packets delivered to a particular node over the period of time is known as throughput of that specific node. Overall network throughput can be obtained by cumulating throughput of all the nodes in the network.

End to End Delay: Time taken between sender dispatching a packet and receiver getting it is described as end to end delay. It can also be described as time taken between packet sent from a specific layer and received at the same layer at the recipient. In our case we take the time taken between application layer of the sender and recipient.

Packet Delivery Ratio (PDR): Is one of the most important and widely used QoS metrics in network communication. PDR can be measured over single and multi-hop communication. However in this study, only single hop broadcast packet delivery ratio is evaluated. Delivering beacons to neighboring vehicles is of utmost importance in VANET because none or limited neighborhood information can lead ineffectiveness of safety applications.

Generally one-hop broadcast PDR can be described in two ways, 1) number of vehicles that successfully receive a broadcast message within the communication range of the sender, PDR-recipients; 2) percentage of beacons received by specific vehicle(s) from a specific sender, PDR-beacons. Most of the previous studies use either one of the two PDR criteria. In this study results are presented based on both PDR criteria. Furthermore more suitable PDR criterion for VANET is also discussed; this is also one of the contributions of this study.

There are no standard values for the measurement of above mentioned metrics; however we assume some logical values based on the results obtained via simulations. Broadly stating, following steps were taken during the course of this study.

- A simulation grid is designed with a 6-lane highway at its centre
- Vehicles are pseudo-uniformly deployed on the highway
- NS-2 built-in 802.11p module’s simulation parameters were appropriately set (see section 4) to match VANET draft standard
- Two sets of simulations were carried out,
 - For first set, CR of all nodes was fixed at maximum while BGI and SB size were changed within practical range limits however other settings remain similar
 - In second set, BGI was fixed at 100ms; on the other hand both CR and SB size were changed within practical range limits and step size however other settings remain similar
- Results for each of the given QoS parameters were extracted from simulation traces.

4 Simulation Setup

Choosing a realistic road scenario and practical settings is important for a meaningful outcome. Following are the traffic scenario and simulation environment settings that are used in this study.

4.1 Traffic Scenario

Figure-1 illustrates the simulation grid. The simulation scenario consists of a six lane 6km long highway with three lanes in each direction. Each lane has a width of 3.66m while lanes on either direction are divided by two meters of separator distance. To limit the boundary effect on results; first, we only analyze communication at the central 2000 meter area, second, the minimum distance between road edges and grid boundary is kept a minimum of $\max CR/2$ i.e. 500m.

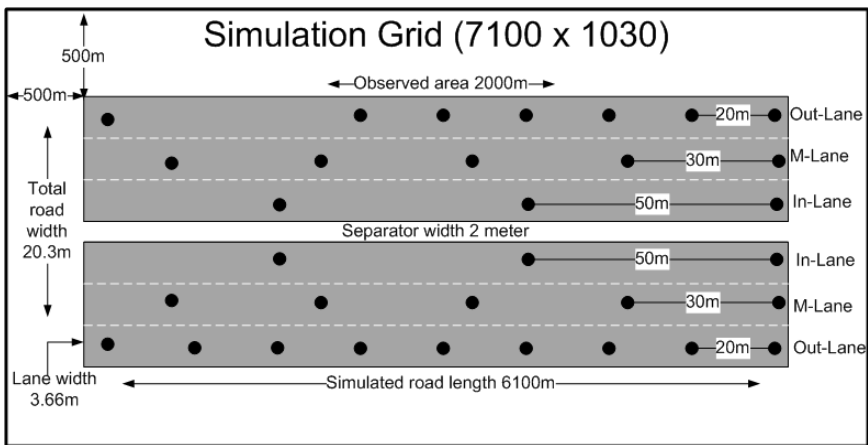


Fig. 1. An illustration of simulation grid and highway settings

A total of 1240 nodes are pseudo uniformly deployed on the highway. A total of 600 nodes (300each) are placed on both of the outer most lanes with a distance of 20 meters in-between. Each of the two innermost lanes has 120 nodes distanced at 50 meters apart. Other two lanes contain 400nodes (200 each) in all with intermediate distance of 30 meters. Distances of 20, 30 and 50 meters depict a minimum safety distance required at the speeds of 40, 60 and 100km/h respectively. In most of the highway situations outermost lanes are populated with slower vehicles and the fastest vehicles travel in innermost lanes which is the concept behind the pseudo uniform placement of nodes.

Vehicle density of 207veh/km provides sufficient number to produce a realistically congested channel environment. Trace distance variable represents the distance with reference to sender, up to which the communication is tracked by ns-2. In order to gain computational efficiency in terms of processing time and storage, we set the trace distance as current Communication Range (CR) + 300m. When using TRG model in ns-2 DSRC module, an addition of 300m to CR ensures coverage of all received/lost beacons.

4.2 Simulation Settings

We use version 2.34 of ns-2 [30] with an overhauled 802.11 PHY and MAC. NS-2 is a good choice considering its credibility among network research community. A survey in [31] reveals ns-2 as the most frequently used simulation tool in VANET papers. Some of the main parameters and their corresponding values used here are shown in Table-1. Brief descriptions of the parameters are given as under.

Communication Range: To obtain desired communication range i.e. 1000m using Two Ray Ground (TRG) model we setup simulation as interference free environment.

Table 1. VANET parameters and their respective settings

Parameter	Corresponding value/s	Parameter	Corresponding value/s
Comm. Range (m)	50, 100, 200...1000	SINR_PreambleCapture	4dB
SB generation interval (ms)	50, 100, 150... 500	SINR_DataCapture	10dB
SB payload size (B)	200, 300 ... 800	Antenna height	1.5m
Frequency	5.885GHz	Antenna gain Gt, Gr	2.512dB
Basic data rate	3Mbps	Slot time	16 μ s
Data rate	6Mbps	SIFS time	32 μ s
Bandwidth	10MHz	Preamble length	32 μ s
Noise floor	-99dBm	PLCP header length	8 μ s
RxTh	-91dBm	Contention window	min. 15 / max. 1023
CSTh	-94dBm	Channel load	various settings
Preamble/Data Capture	On	Simulation time	21sec/each

TRG propagation model in NS2 calculates the distance according to eq.1 if the distance is less than cross-over distance. For distances greater than crossover, calculation is done using Freespace model (eq.2)

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \quad \text{if } d \leq d_c \quad (1)$$

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L} \quad \text{if } d > d_c \quad (2)$$

Where P_r , P_t are power received and power transmitted, h_t , h_r are the transmitter and receiver antenna heights, G_t , G_r are antenna gain at transmitter, receiver, λ is the frequency wavelength and L is system loss. Crossover distance is calculated as eq. 3.

$$d_c = \frac{4\pi h_t h_r}{\lambda} \quad (3)$$

Various CRs were used as shown in Table-1. Obtained Communication and Carrier Sense ranges with their respective power values are shown in Figure-2.

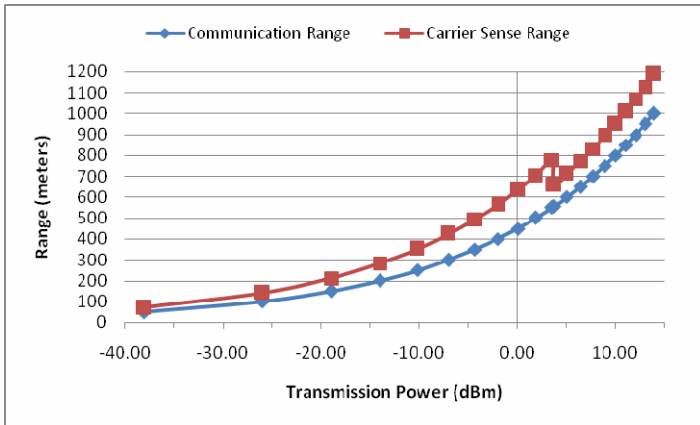


Fig. 2. Obtained transmission power values using Two Ray Ground propagation model

Frequency: Although FCC has allocated a 75MHz Dedicated Short Range Communication (DSRC) frequency spectrum of 5.850-5.925 for VANET but only a single channel of 5.885GHz onwards is allocated for periodic safety beacons which we have used in our simulations.

Basic data rate/Data rate: Date rate of 3Mbps is the minimum supported by 802.11p thus it is used as basic data transfer rate to transmit preamble. According to a recent study [32] 6Mbps is the optimal date rate for heterogeneous VANET environment.

Bandwidth: Channel bandwidth of 10MHz is analogous to the control channel bandwidth of DSRC spectrum as specified by FCC in USA.

Noise floor: For a bandwidth of 10MHz, default noise floor value of -99dBm is used.

Reception Threshold (RxTh): is the minimum power required by a receiver to successfully decode the message. RxTh can be calculated using eq.4.

$$\text{RxTh} = \text{Receiver Noise floor} + \text{SNR} \quad (4)$$

In DSRC, to successfully receive a frame within 10MHz channel and 6Mbps data rate, a Signal to Noise ratio of 8dB is required [32]. Thus an ultimate choice for RxTh = $-99 + 8 = 91(\text{dBm})$.

Carrier Sense Threshold (CSth): Carrier sense range is the range up to which a receiver is able to sense ongoing communication but is unable to decode it successfully. The threshold value for this is obtained from the latest updated settings of default ns-2 802.11p module [33].

Preamble/Data Capture: When set to ON it enables a receiver to differentiate between frame header and payload. It further enables a receiver to choose the strongest frame header among several. It is also well known to enhance packet reception rate. The corresponding parameters of SINR_PreambleCapture and SINR_DataCapture are set to default values of 4dB and 10dB respectively.

Antenna Gains (Gt, Gr): Default ns-2 antenna height is 1.5m, and transmitter/receiver antenna gain of 2.512 is similar to that of [34].

All the MAC layer settings (Slot time, Short Interframe Space (SIFS), Preamble length, PLCP header length, Contention window size are unchanged from default except RTC/CTS which is chosen to be 3000 to effectively disable it. Channel load in the observed area varies depending on the communication range settings. A total of 139 simulations were carried out each with a 21sec simulation time. Simulation results for first second are truncated to observe steady network conditions. Overall more than 730GB of ns-2 trace data was generated.

5 Results and Analysis

5.1 Per-node Throughput Results

Figure-3 shows the impact Safety Beacon (SB) Generation Interval and SB size on per-node throughput. As shown, if interval is below 200msec, varying the packet size does not bring any significant change. Noticeable variations occur with BGI of 200msec and above as throughput increases significantly with increment in SB size. With SB size of 700-800B throughput increases as the BGI is increased. Maximum throughput is achieved with 800B SB size and 500sec of BGI. For maximum throughput, safety application with stringent delay requirements but requiring lesser

amount of information to be transferred can chose SB interval between 200-300msec, while safety applications requiring larger information exchange and lesser urgency can use 400-500msec SB interval.

To monitor the effect of Communication Range (CR) on per-node throughput we fix SB interval to 100ms and vary the CR and SB size. Results in Figure-4 show that higher throughput is achievable with wider CR. Furthermore with communication range of 500m and below larger SB size is better suited for maximum output while overall process is reversed with CR between 500-1000m. Thus it is desirable to have maximum CR along minimum SB size for maximum throughput.

Comparison of Figure-3 and Figure-4 reveals that SB interval is most significant among parameters used here in terms of per-node throughput control. Furthermore it is clearly shown that techniques relying on only reducing communication range/power will have negative impact on overall throughput. Nevertheless, an ideal combination for maximum throughput in the given scenario would be 1000m CR, 800B SB size and 500msec BGI interval.

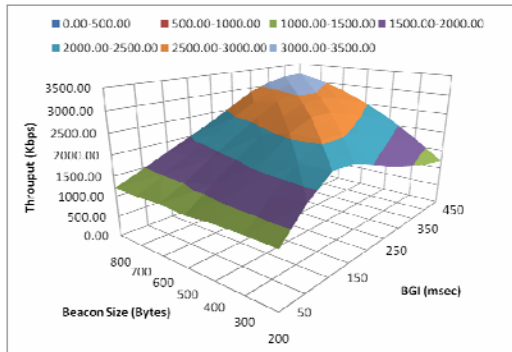


Fig. 3. Throughput results for BGI vs Beacon size, (CR=1000m)

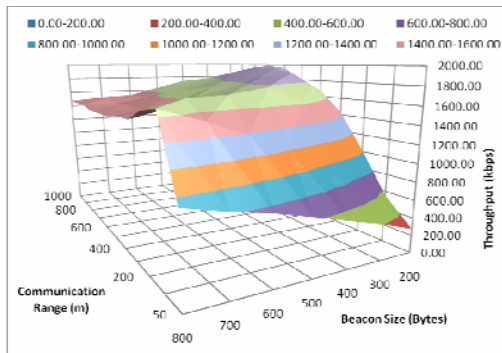


Fig. 4. Throughput results for CR vs Beacon size, (BGI=100msec)

5.2 End-to-end Delay (e2e delay) Results

Unlike many studies that estimate e2e delay in non-interfering environment, we calculate averages of several nodes in a fully deployed network. Although graphs obtained are not smooth in nature however the method applied is useful in determining overall trends of e2e delay within the boundaries of studied parameters.

It is clearly visible from the Figure-5 and Figure-6 that a smaller SB size is desirable for minimum e2e delay over larger distances. Moreover e2e delay with 800B size over longer distances is still within the acceptable limits (10-20msec) and does not require increment in BGI. As of the results obtained with BGI interval of 50msec (not shown here for presentation reasons), e2e delay varies from 89msec to 570msec for SB sizes of 700 to 800B respectively. Thus it is safe to conclude that BGI of 50msec and below is not suitable with larger SB size under stringent e2e delay requirements.

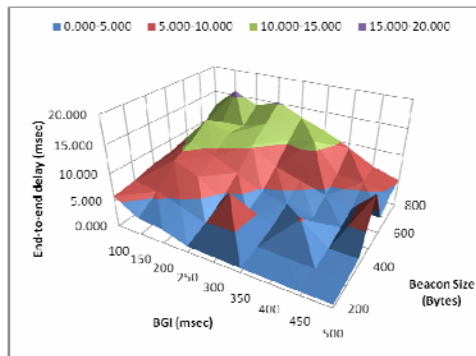


Fig. 5. e2e delay results for BGI vs Beacon size, (CR=1000m)

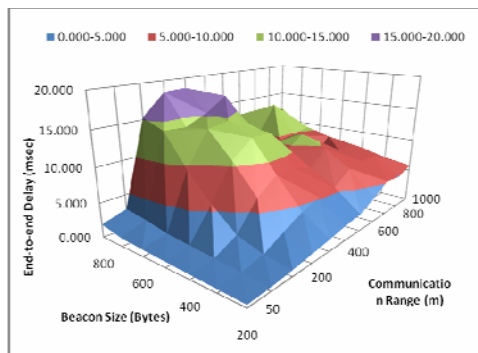


Fig. 6. e2e delay results for CR vs Beacon size, (BGI=100msec)

5.3 Packet Delivery Ratio (PDR) Results:

PDR can be obtained in two ways, either by calculating percentage of recipients of broadcast packet (PDR-recipients) or by calculating percentage of packets successfully received by receiving nodes from a specific sender (PDR-beacons).

PDR-beacons results in Figure-7 and Figure-8 show that although reducing CR improves PDR-beacons but the variation is significant only in shorter range i.e. 200m or less. However by carefully adjusting the BGI it is possible to achieve higher PDR-beacons e.g. increasing 50msec BGI almost doubles the delivery rate at CR of 1000m. It is also evident from results that smaller SB size contributes towards higher PDR-beacons but it is of lesser importance when compared with BGI and CR.

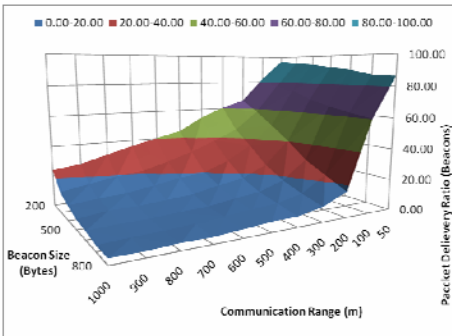


Fig. 7. PDR-beacon results for CR vs Beacon size, (BGI=100msec)

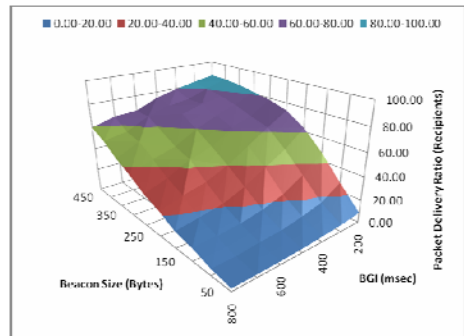


Fig. 8. PDR-recipient results for BGI vs Beacon size, CR=1000m

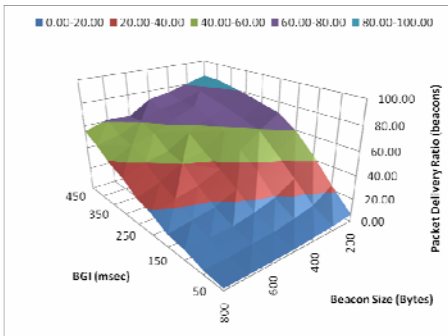


Fig. 9. PDR-beacon results for BGI vs Beacon size, (CR=1000m)

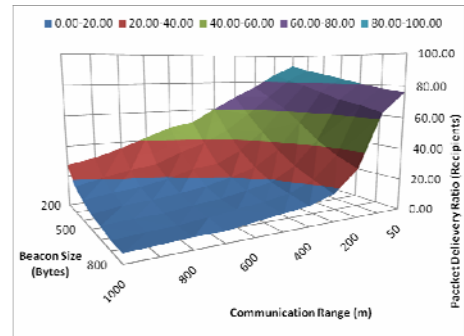


Fig. 10. PDR-recipient results for CR vs Beacon size, (BGI=100msec)

PDR-recipients results in Figure-9 and Figure-10 show similar behavior as of PDR-beacons. The only notable difference is that with similar parameter settings, rate of PDR-recipients is slightly higher than PDR-beacons at almost all times. The logical explanation could be that while taking average number of recipients, nodes with only few receptions may also be included in calculations, which is unrealistic. While

averaging PDR-beacons also includes nodes with fewer receptions, nonetheless it shows more practical statistical values than PDR-recipients. Hence it is concluded that more reliable results are obtained when PDR-beacons is used as the QoS parameter.

6 Conclusion and Future Work

In this paper simulation based performance evaluation single-hop periodic beacons is presented. For safety communication it was observed that optimum PDR at maximum CR (1000m) can be achieved at maximal BGI i.e. 500ms. If the BGI interval is fixed at 100ms the CR needs to be reduced significantly (e.g. < 200m) for higher PDR. However for non-safety applications (e.g. multimedia advertisements) per-node throughput may serve as a better QoS metric. Choosing optimal CR and BGI for higher per-node throughput depends on the message size e.g. for the given scenario, with a message size of 800B ideal CR, BGI are 1000m and 500ms respectively. In the light of obtained results it is understood that situation-aware dynamic adaption of CR and BGI is essential for beaconing optimization in VANETs. Furthermore a comparative analysis between PDR-beacons and PDR-recipients shows that the later is a more realistic choice for reliability reasons. In the future simulations are to be performed with a more realistic propagation model i.e. Nakagami.

References

1. Abdalla, G.M.T., Abu-Rgheff, M.A., Senouci, S.M.: Current trends in vehicular ad hoc networks. In: IEEE Global Information Infrastructure Symposium (GIIS), Marrakech, Morocco (July 2007)
2. Jiang, D., Delgrossi, L.: IEEE 802.11p: Towards an international standard for wireless access in vehicular environments. In: IEEE Vehicular Technology Conference (VTC), pp. 2036–2040 (May 11-14, 2008)
3. Vehicle Safety Communications Project, Task 3 Final Report, Identify intelligent vehicle safety applications enabled by DSRC (March 2005), <http://www.intellidriveusa.org/documents/vehicle-safety.pdf>
4. Jiang, D., Taliwal, V., Meier, A., Holfelder, W., Herrtwich, R.: Design of 5.9 GHz DSRC-based vehicular safety communication. IEEE Wireless Communications 13(5), 36–43 (2006)
5. Moustafa, H., Zhang, Y.: Vehicular networks techniques, standards and applications. Taylor & Francis Group, USA (2009)
6. Mittag, J., Schmidt-Eisenlohr, F., Killat, M., Harri, J., Hartenstein, H.: Analysis and design of effective and low-overhead transmission power control for VANETs. In: ACM, VANET 2008, San Francisco, California, USA (September 2008)
7. Network on Wheels Project (NOW), <http://www.network-on-wheels.de>
8. Williams, B., Camp, T.: Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks. In: ACM MOBIHOC (2002)
9. Lou, W., Wu, J.: Double-Covered Broadcast (DCB): A Simple Reliable Broadcast Algorithm in MANETs. In: INFOCOM (2004)

10. Lipman, J., Boustead, P., Chicharo, J.: Reliable optimized flooding in ad hoc networks. In: IEEE 6th CAS Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication (May 2004)
11. Alshaer, H., Horlait, E.: An optimized adaptive broadcast scheme for Inter-vehicle communication. In: Vehicular Technology Conference, VTC (2005)
12. Wischhof, L., Rohling, H.: Congestion control in vehicular ad hoc networks. In: IEEE International Conference on Vehicular Electronics and Safety, pp. 58–63 (October 2005)
13. Zang, Y., Stibor, L., Cheng, X., Reumerman, H.-J., Paruzel, A., Barroso, A.: Congestion control in wireless networks for vehicular safety applications. In: The 8th European Wireless Conference, Paris, France, p. 7 (2007)
14. Artimy, M.M., Robertson, W., Phillips, W.J.: Assignment of dynamic transmission range based on estimation of vehicle density. In: ACM, VANET 2005, Cologne, Germany (September 2005)
15. Torrent-Moreno, M., Santi, P., Hartenstein, H.: Fair sharing of bandwidth in VANETs. In: ACM, VANET 2005, Cologne, Germany (September 2005)
16. Torrent-Moreno, M., Santi, P., Hartenstein, H.: Distributed fair transmit power adjustment for vehicular ad hoc networks. In: 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, SECON 2006, vol. 2, pp. 479–488 (September 2006)
17. Torrent-Moreno, M., Mittag, J., Santi, P., Hartenstein, H.: Vehicle-to-Vehicle communication: fair transmit power control for safety-critical information. IEEE, Los Alamitos (2009)
18. Jiang, D., Chen, Q., Delgrossi, L.: Communication Density: A Channel Load Metric for Vehicular Communications Research. In: IEEE International Conference on Mobile Adhoc and Sensor Systems MASS, October 8-11, 2007, pp. 1–8 (2007)
19. Li, X., Nguyen, T., Martin, R.: Analytic Model Predicting the Optimal Range for Maximizing 1-hop Broadcast Coverage in Dense Wireless Networks. In: ADHOC-NOW, Canada (2004)
20. Li, X., Nguyen, T., Martin, R.: Using Adaptive Range Control to Maximize 1-Hop Broadcast Coverage in Dense Wireless Networks. In: IEEE SECON, Santa Clara, CA (2004)
21. Chigan, C., Li, J.: A delay-bounded dynamic interactive power control algorithm for VANETs. In: IEEE International Conference on Communications ICC 2007, pp. 5849–5855 (June 2007)
22. Torrent-Moreno, M., Jiang, D., Hartenstein, H.: Broadcast Reception Rates and Effects of Priority Access in 802.11-Based Vehicular Ad-Hoc Networks. In: ACM VANET (2004)
23. Xu, Q., Mak, T., Ko, J., Sengupta, R.: Vehicle-to-Vehicle Safety Messaging in DSRC. In: ACM VANET (2004)
24. Chen, Q., Schmidt-Eisenlohr, F., Jiang, D., Torrent-Moreno, M., Delgrossi, L., Hartenstein, H.: Overhaul of IEEE 802.11 modeling and simulation in ns-2. In: Proc. Of 10th International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM 2007, Chania, Crete Island, Greece (October 2007)
25. Khan, A., Sadhu, S., Yeleswarapu, M.: A comparative analysis of DSRC and 802.11 over Vehicular Ad hoc Networks. Dept. of Computer Science, University of California (2008)
26. Yousefi, S., Bastani, S., Fathy, M.: On the Performance of Safety Message Dissemination in Vehicular Ad Hoc Networks. In: Universal Fourth European Conference on Multiservice Networks. ECUMN 2007, February 14-16, pp. 377–390 (2007)
27. Raya, M., Hubaux, J.: The security of vehicular ad hoc networks. In: Proc. 3rd ACM Workshop SASN, Alexandria, VA, pp. 11–21 (November 2005)

28. Mughal, B.M., Wagan, A.A., Hasbullah, H.: Efficient congestion control in VANET for safety messaging. In: International Symposium in Information Technology (ITSim), vol. 2, pp. 654–659 (June 2010)
29. Olson, P.: Perception-response time to unexpected roadway hazards. *Human Factors* 28(1), 91–96 (1986)
30. Network Simulator – ns-2, <http://www.isi.edu/nsnam/ns/>
31. Olariu, S., Weigle, M.C.: *Vehicular Networks: From Theory to Practice*. 13.2. Chapman & Hall/CRC, Boca Raton (2009)
32. Jiang, D., Chen, Q., Delgrossi, L.: Optimal Data Rate Selection for Vehicle Safety Communications. In: Proceedings of the fifth ACM international workshop on Vehicular Inter-NETworking, VANET 2008. ACM, New York (2008)
33. Overhaul of IEEE 802.11 Modeling and Simulation in NS-2, http://dsn.tm.uni-karlsruhe.de/english/Overhaul_NS-2.php
34. Yang, L., Guo, J., Wu, Y.: Channel Adaptive One Hop Broadcasting for VANETs. In: 11th International IEEE Conference on Intelligent Transportation Systems, ITSC., October 12-15, pp. 369–374 (2008)

Analysis of Routing Protocols in Vehicular Ad Hoc Network Applications

Mojtaba Asgari^{1,2}, Kasmiran Jumari¹, and Mahamod Ismail¹

¹ Department of Electrical, Electronic & Systems Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia
`{m.asgari,kbj,mahamod}@eng.ukm.my`

² Department of Computer Engineering, Faculty of Engineering, Islamic Azad University, Mashhad Branch, Mashhad, Iran
`m.asgari@mshdiau.ac.ir`

Abstract. Vehicular Ad Hoc Networks (VANETs) are self-organizing, self-healing networks that provide wireless communication among vehicles and roadside equipment. Providing safety and comfort for drivers and passengers is a promising goal of these networks. Designing an appropriate routing protocol according to the network application is one of the essential requirements for implementing a successful vehicular network. In this paper, we report the results of a study on routing protocols related to vehicular applications and their communication needs. In general, all VANET communications can be implemented by either unicast or multicast routing protocols. The results of the study showed that multicast protocols, including geocast and mobility-based routing, are more promising than others for fulfilling the application requirements and, consequently, more research of these protocols is needed.

Keywords: vehicular networks, VANET applications, VANET, ad hoc networks, MANET, routing.

1 Introduction

Every year, millions of road accidents occur in the world. This result in the loss of more than 1.2 million lives and destruction of about 1 to 2% of total of gross national products [1]. Annual averages from 1980 to 2008 in U.S. shows 42,810 people killed and 2.928 million wounded [2], at an estimated economic cost of over 231 billion dollars; more than 40,000 people die and 1.8 million people are injured on European roads each year [3]. Furthermore, traffic jams and stop lights, in large cities create a massive waste of time and fuel, especially during rush hours. Providing a reliable vehicular communication system to transfer and share relevant information with drivers and vehicles can significantly increase road safety and improve traffic distribution on congested roads.

Vehicular Ad Hoc Network (VANET) is a rising subclass of Mobile Ad Hoc Networks (MANETs) which provide wireless communication between mobile nodes (vehicles) and vehicles to roadside equipments. The efficiency of this network depends on how quickly and accurately routing protocols can make decisions to route

data. Large-scale and highly dynamic topology, frequently partitioned and disconnected network, and patterned mobility of the nodes are some challenging characteristics of VANET which result in significant loss rates and very short communication times [4]. These attributes influence the performance of routing solutions suitable for ordinary MANET. Hence in recent years, researchers have attempted to overcome these problems and propose some new protocols for vehicular networks.

Previous survey studies on VANET routing protocols have been conducted by [4-9]. This study extended the previous studies because it focused on classifying routing protocols based on specific vehicular applications and their requirements to determine an appropriate routing strategy for each application.

2 Vehicular Network Applications

In this section, we present an overview of applications formed by vehicular ad hoc networks, then discuss their requirements with respect to routing, in next section. A good overview of these applications is available in [5, 10-15].

In Figure 1, we demonstrate the different classifications of vehicular ad hoc network applications. VANET applications consist of two main categories: safety and user (non-safety) [12]. Safety applications are the primary purpose of vehicular communication technology development. They increase traffic safety and significantly reduce the number of road accidents by providing warning messages for drivers a few moments before collision. User applications provide information, advertisement and entertainment services for drivers and passengers. Each category can also be divided into several classes. Safety applications are comprised of public safety, traffic management, and traffic coordination and driver assistance applications. User applications include traveler information support and comfort applications [10].

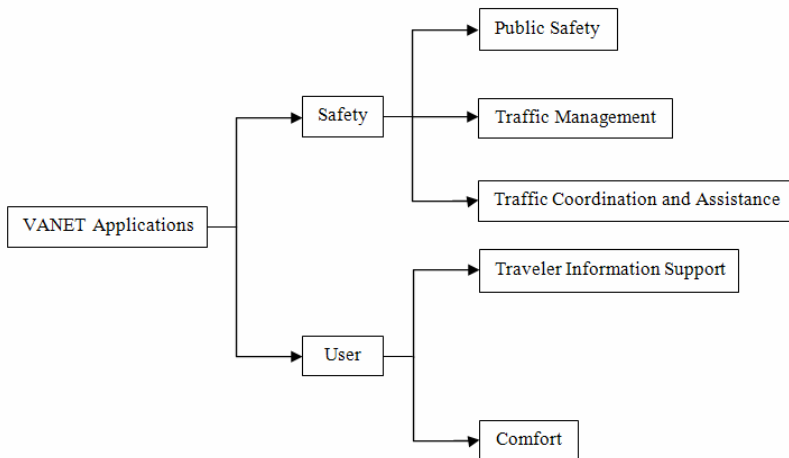


Fig. 1. Vehicular ad hoc network applications

2.1 Safety Applications

Public Safety Applications. This domain primarily refers to applications and systems that try to prevent or lower the number of accidents. Their goal is to protect vehicles and their occupants from damage by accidents.

Cooperative Collision Avoidance. A collision avoidance system would continuously monitor the vehicle's surroundings (distance to other vehicles) and alert the driver when another vehicle could cause a collision. If necessary, the system can respond with an automatic action (e.g. braking) to prevent an imminent collision [16]. Even if the collision is unavoidable, special safety arrangements can be made to prepare vehicles for accidents (tighten seat belts, pump up air bags, etc.) [10].

Cooperative Collision Warning. Scientists have proposed collision warning systems (also called emergency event warning systems) to reduce the number of vehicle collisions. Vehicles that detect an unexpected event like an accident or a driver on the wrong side of the road could broadcast a warning message to all vehicles in their surrounding area (also known as the Zone of Relevance (ZOR)). This would notify all drivers in the ZOR in advance [16]. These systems also may take post-collision conditions into consideration and inform vehicles near the accident [11].

Traffic Management Applications. Traffic management is another application of VANETs which works to adjust traffic by exchanging messages between vehicles and/or roadside equipments. This avoids traffic jams and their resultant accidents, reducing travel time and fuel consumption. Cooperative road traffic monitoring, traffic light scheduling, vehicle tracking, and emergency vehicle signals are some application examples of this type.

Cooperative Road Traffic Monitoring. This system provides accurate and localized traffic information of a ZOR for a vehicle. The ZOR can be several kilometers around the vehicle, and the information can be disseminated via vehicle-to-vehicle communication in a multi-hop manner. This information is useful in situations such as rerouting and estimating the time to a destination [10].

Traffic Light Scheduling. In this application, vehicles communicate with traffic lights to achieve an optimal schedule. This communication can provide further information which can increase the scheduling efficiency, such as the number of vehicles queued up at a traffic light and vehicles estimated to reach it soon [4].

Vehicle Tracking. In vehicle tracking systems, a vehicle equipped with a GPS can periodically send its location information (coordinates) to a computer at a data center via network technologies. Then, the information can be seen on digital maps via the Internet or specialized software, allowing the vehicle's movements to be followed. This system can be used as a dependable car alarm system to determine whether the vehicle has been stolen. In addition, in transportation or public transit systems, customers can, for example, use the vehicle tracking system to track their cargo or to check the current position of the nearest taxi through the Internet [17].

Emergency Vehicle Signals. The main purpose of emergency vehicle warning signals is to reduce the travel time of emergency vehicles (e.g., ambulances, fire engines, and

police cars). The faster an emergency vehicle can get to the scene of an emergency, the higher the probability that lives can be saved [18]. When an emergency vehicle is traveling on a roadway, it sends messages to other vehicles so the drivers can clear the way. The message includes extensive information, including the emergency vehicle's current position, lane, speed, destination, and intended route. The vehicle that receives the warning message can also forward it to other vehicles in the path to give the drivers more time to respond. In addition, an emergency vehicle can send messages to traffic lights in a route to set them to green just before the emergency vehicle reaches the intersection [18, 19].

Traffic Coordination and Driver Assistance Applications. In such applications, vehicles work together to coordinate their motions with each other and to also assist drivers to maneuver safely in actions such as passing and changing lanes. A driverless vehicular system is certainly the ultimate objective of the applications [20].

Platooning. The vehicle following or platooning system allows a group of vehicles on highways to follow each other to form a convoy so that they can travel closely and safely [10, 11]. The front vehicle, which is the lead vehicle, periodically sends messages to the group to set the average speed of the convoy. Each driver (or self-driving vehicle) tracks the preceding vehicle closely by accelerating or decelerating the vehicle, according to the information received [21]. Such a system could lead to a higher safety level for passengers due to the vehicles' continuous monitoring of the condition of the road [11]. Platooning also has the potential to increase the throughput of vehicles on highways since the vehicles can travel closer together at higher speeds [10, 21].

Passing and Lane Change Assistance. Passing and changing lanes are often the source of serious lateral accidents because nearby vehicles may be located in the driver's blind spot [10, 21]. In such situations, a lane-change assistance system that has information about the locations of nearby vehicles (e.g., the distance to vehicles in adjacent lanes and their relative speed) can assist a driver by providing a warning when a lane change is unsafe due to the presence of other vehicles [22].

2.2 User Applications

Traveler Information Support. These applications are used to provide updated local information and warnings about road conditions for drivers.

Local Information and Advertisements. In this application, a driver could get, for example, local maps, information, or advertisements of restaurants, gas stations, hotels, and parking places when the vehicle approaches specific locations (hot spots). This information can be provided by messages that are broadcasted periodically by access points in the hot spots or by having other vehicles relay the messages [17].

Warnings About Hazardous Road Conditions. This system is designed to notify drivers about hazardous road conditions (e.g., ice, oil, and bumps). A vehicle equipped with sensors can collect data about the road conditions, process the data, identify risky situations, and warn the driver (e.g., about the maximum safe speed on the road). The warning is also sent to other vehicles in the ZOR to inform the drivers about the situation [18].

Comfort. These applications can make travel more enjoyable for drivers and passengers by providing value-added services, such as Internet connectivity and direct communication between two vehicles. Compared to other classes, these services are multi-hop, end-to-end communication, and it is the user's choice whether to use them or not. Hence, their priority is relatively low [19].

Peer-to-Peer. This service allows communication between two vehicles via other vehicles in the network without the need for any application server. Live voice communication, video/voice communication, and instant messaging between drivers and passengers in separate vehicles are some applications of this type [11, 17].

Internet Connectivity. Providing Internet access to the occupants of vehicles, where other wireless Internet technology options (e.g., WiMAX and WiFi) are not available, is a valuable capability because it can provide a wide range of services, including email, web browsing, file transfer, and voice over IP. Even if a vehicle is connected to the Internet via an access point at a hot spot, its occupants can share this connectivity and serve other vehicles through VANET [11].

Drive-through Toll/Park Payment. Although many non-standard systems exist for this application and work well, drive-through toll/park payment capability allows fast and convenient payment transactions without the driver's having to change the vehicle's normal operating speed [22].

3 Communication Requirements of Vehicular Applications

In this section, we report the results of our study of the main communication requirements of vehicular ad hoc applications regarding routing. Although such studies should cover all aspects of the requirements for their effective use, we only considered the most prominent requirements in our study. Knowledge of both applications and requirements is important for the design and analysis of efficient routing solutions according to the targeted applications. Thus, we assessed the requirements of the applications introduced in the previous section, and these requirements are summarized in Table 1. The information included in Table 1 is described below.

Implementation Type. Vehicular communication (VC) systems are classified according to their implementation requirements into inter-vehicle communication (IVC) (also called vehicle-to-vehicle (V2V) communication) and roadside-to-vehicle communication (RVC) (also called infrastructure-to-vehicle (I2V) and vehicle-to-infrastructure (V2I)). IVC systems are used in applications in which vehicles want to exchange data, whereas RVC systems are used in case the information must be transmitted between roadside units (RSUs) and the vehicles.

IVC is an ad hoc, infrastructureless communication model, without any assistance of RSUs; it only requires some on-board units (OBUs), also called in-vehicle equipment (IVE), to exchange information among vehicles. In this case, depending on

the distance between the vehicles, communication can occur as single-hop or multi-hop IVC (SIVC or MIVC) [10]. Applications such as lane-change assistance, which require short-range communication, use SIVC systems. MIVC systems are used in applications that require long-range communication, such as monitoring road traffic and Internet connectivity. An MIVC system is more complex than an SIVC system and requires multi-hop routing. For example, connectivity of the nodes in a network that has a low density of vehicles is not guaranteed, and data may be lost. Therefore, the appropriate design of multi-hop routing protocols requires further consideration.

In RVC systems, RSUs communicate with OBUs. Depending on the application, such systems can be divided into sparse RVC (SRVC) systems and ubiquitous RVC (URVC) systems [10]. SRVC systems are useful for application services at hot spots, e.g., scheduling traffic lights, a business that wishes to inform travelers of its location and prices, and drive-through payment for parking at an airport. These systems can be combined with IVC systems to provide longer transmission ranges. A URVC system is intended for providing high-speed communication across all roads via RSUs. By having such a system, all vehicular applications can be designed, but it is very difficult to achieve that due to the considerable costs for full coverage of all roads, especially in large countries.

Communication Type. In vehicular networks, communication types are similar to one-to-one, one-to-many, and one-to-all in conventional networks, which have been adapted for vehicular communications. In one-to-one communications (unicast), messages are sent from one vehicle to another specific vehicle via MIVC systems. Such communications usually require a long period of connectivity between the vehicles to operate effectively, but, in most cases, vehicular networks are highly dynamic networks with intermittent connectivity. Furthermore, few VANET applications (e.g., vehicle tracking and peer-to-peer) require one-to-one communications.

The behavior of one-to-many (multicast) and one-to-all (broadcast) protocols are different; the former sends a message to multiple vehicles based on a particular group definition, while, in the latter, the message is disseminated to all vehicles. In each vehicle, the MAC layer (IEEE 802.11p standard) normally does single-hop broadcast by disseminating a message to the vehicles in its transmission range. Each time a vehicle receives a broadcast message, it is stored and immediately rebroadcasts it to the neighbors. This procedure could lead to message flooding (broadcast storm problem) [23], especially in high traffic density scenarios, and make a non-scalable network. In vehicular applications, selective broadcast is more applicable than full broadcast. Selective broadcast, which is a kind of multicast protocol, is an optimized broadcast mechanism with a reduced area for involving vehicles. A fleet of taxis that relays messages among themselves is an example of a multicast system, whereas an ambulance alarm message, which must be conveyed to all nearby vehicles so they can pull over quickly and safely, is a selective broadcast system [24].

Table 1. VANET application requirements regarding routing (***- extremely needed, **- needed, *- not needed)

Application		Implementation type	Communication Type	Transmission Mode	Real-time requirement
Public Safety	Cooperative collision avoidance	SIVC or MIVC (highway) ----- SRVC (intersection)	One-to-many	Periodic	***
	Cooperative collision warning	MIVC (highway) ----- SIVC or MIVC or SRVC(intersection)	One-to-many	Event-driven	***
Traffic management	Cooperative road traffic monitoring	MIVC	One-to-many	Periodic	**
	Traffic light Scheduling	SRVC	One-to-many	Periodic	**
	Vehicle tracking	SRVC	One-to-one	Periodic	**
	Emergency vehicle signals	MIVC (other vehicles) ----- SRVC (traffic lights)	One-to-many ----- One-to-one	Event-driven	***
Traffic coordination and driver assistance	Platooning	SIVC or MIVC	One-to-many	Periodic	***
	Passing and lane change assistance	SIVC	One-to-many	Periodic	***
Traveler information <small>summary</small>	Local information and advertisements	SRVC or MIVC	One-to-many	Periodic	*
	Road conditions warning	MIVC	One-to-many	Event-driven	**
Comfort	Peer-to-peer	MIVC	One-to-one	Normal	*
	Internet connectivity	MIVC	One-to-one	Normal	*
	Drive-through toll/park payment	SRVC	One-to-one	Normal	*

Transmission Mode. All VANET applications, depending on their functionality, get data from other vehicles, sensors, or both. Each application processes the data and then sends appropriate messages to adjacent vehicles or to roadside equipment. Safety applications send messages to alert people in nearby vehicles to an unsafe condition. Messages that are sent periodically to nearby vehicles to make them aware of a vehicle's current status (e.g., location, speed, and direction) in order to avoid accidents are called periodic messages or beacons. They may also be used in other non-safety applications, such as to provide advertisement data. Event-driven or alert messages that contain the location of the vehicle, the time, and the event type are emergency warning messages sent to other vehicles upon detection of an unsafe condition. These messages have very high priority to make sure that all vehicles receive the message quickly and accurately. In some applications, such as Internet connectivity service, there is no need for any periodic or event-driven messages. These applications are transmitted based on normal transmission mode in conventional networks.

Real-time Requirement. Minimum latency is an important factor for vehicular safety applications. Real-time safety information must be delivered as soon as possible to vehicles that need the information to avoid injuries to passengers or damage to the vehicles. IEEE 802.11p, which was introduced by the IEEE group as a MAC-layer standard for vehicular communications, also defined the minimum allowable latency as 200 μ sec. In general, end-to-end delay is one of the important performance metrics in all communication protocols.

4 Application-Based Classification of Routing Protocols

Table 1 indicates that few VANET applications require unicast routing protocols to communicate. In most cases, a sender sends messages to a group of receivers located behind the sender, such as those encounter a specific hazardous event, or in front of it such as those could transmit the road information, or in a given geographic area, for instance near to a designated spot such as a rainy area, or those can provide a particular service such as fleet of taxis. Such applications are assimilated into multicast communications. In the following, we discuss each of the communication protocols and their types. Also, Table 2 lists routing protocols according to the types.

4.1 Unicast Routing Protocols

Unicast protocols depend principally on either the network topology (such as destination-sequenced, distance-vector (DSDV) routing [25] and ad hoc on-demand distance vector (AODV) routing [25]) or geographical position of the vehicles (such as greedy-perimeter stateless routing (GPSR) [4] and connectivity-aware routing (CAR) [4]).

Topological Routing. Topology-based routing algorithms only count the network topology (including nodes and communication links) to find the routes of the messages. Topology-based schemes, which are mostly associated with MANET routing, use a proactive or reactive approach to create routes. In the proactive

approach (also called the table-driven approach), e.g., DSDV, optimized link state routing (OLSR) [25], and fisheye state routing (FSR) [25], all nodes send periodic routing messages to create or update routing tables, even if there are no traffic data. These protocols require excessive control messages and consume additional bandwidth. On the contrary, reactive or on-demand protocols, such as AODV, temporally-ordered routing algorithm (TORA) [25], and dynamic source routing (DSR) [25], create routes only when they are needed to send data to a destination. These protocols use flooding, and there is a delay to find routes before data can be sent. In highly-dynamic vehicular networks, both protocols are inefficient, because the construction and maintenance overhead of the routes increase as network dynamics increases [26].

Several improvements have been proposed for these protocols to support highly-dynamic vehicular networks. Fast OLSR [27] tries to bear the network dynamic by adapting the frequency of periodic messages. In earlier work [28], it was proposed that additional geographical information be provided for the route request packets of the AODV routing protocol. All such enhancements still are considered as topological routing with some improvement in routing performance.

Position-based Routing. Geographical position-based protocols do not establish the route between the source and the destination. On-the-fly forwarding decisions are based on the position of the destination and the positions of neighbors. These protocols consist of three main components i.e., beaconing, location service, and forwarding strategies [9, 38].

Nodes use beaconing to broadcast short messages (beacons) periodically to inform neighbors of their unique ID and current position. Neighbors use the information they receive to update their location tables. In this way, information related to next-hop neighbors is obtained from the beacons. Information about the position of a destination can be obtained from a location management service or by flooding messages in the anticipated destination area to find the desired node and get its reply. The requirement associated with forwarding methods is to forward messages from a source to a destination effectively.

There are three kinds of forwarding in vehicular position-based routing protocols: greedy forwarding, directional flooding [34, 38], and hierarchical forwarding [34]. In greedy forwarding (e.g., GPSR, greedy perimeter coordinator routing (GPCR) [4], GVGrid [39], anchor-based street and traffic aware routing (A-STAR) [4, 40], and vehicle-assisted data delivery (VADD) [4, 41]), after a node has determined the location of the destination, the closest neighbor to the destination is selected as a next hop, and messages are sent to this neighbor. The algorithm is repeated at the nodes within the forwarding path until the message reaches the destination. On the other hand, directional flooding (such as the mobility-centric data dissemination algorithm for vehicular networks (MDDV) [33, 42]) disseminates messages to the nodes located in the same direction as the destination by flooding. Recipients rebroadcast the messages if they are situated in the geographical area specified by the information in the message. In this case, improved broadcasting algorithms can be used to avoid broadcast storms and minimize overhead. In hierarchical forwarding (e.g., GeoGRID [43]), the network is divided into several clusters. A cluster is a set of nodes that have some common characteristics (such as direction and speed) and are stable for a given period of time. Clusters forward messages to each other via gateway nodes. Several levels of the clusters can be defined.

Table 2. Routing protocols according to the kind of communication in VANET applications

Unicast		Multicast	
Topological	Position-based	Geocast	Mobility
AODV	GPSR (manet)	DRG [35]	LBF [37]
CBRP [25]	CAR	GAMER	MDDV
DSR	ACAR [29]	GeoGRID	OABS
OLSR	DREAM [25]	IVG [36]	ODAM [24]
Fast OLSR	GPCR	LBM	RBM [36]
HSR [25]	GSR [4]	GHM [36]	SB [24]
DSDV	LAR (manet)[25]	PBM [36]	SOTIS [10]
FSR	LORA-CBF [21]	SPBM [36]	VTRADE [20]
TORA	MDDV	RSGM [36]	UMB [24]
LMR [25]	MORA [30]		VADD
WRP [25]	MURU [31]		ROMGSP
	VADD		
	SAR [18]		
	A-STAR		
	STAR (manet)[25]		
	GeoOpps [4]		
	MaxProp [32]		
	PDGR [33]		
	GyTAR [4, 34]		
	GRANT [4]		
	PBR [26]		
	GVGrid		
	GeoGRID		

In position-based routing protocols, when network dynamics increase, determining the positions of the destination and the neighbors becomes unstable. As a result, the destination position is not valid when the message reaches that point. To deal with this problem, the defined geographical area of the destination can be increased, but this also leads to an increased number of nodes in that area and bandwidth waste. However, some studies show that position-based routing is more appropriate than topological routing in highly-dynamic vehicular networks.

4.2 Multicast Routing Protocols

Multicast routing protocols consist of two classes: geocast protocols and mobility-based protocols.

Geocast Routing. The geocast approach is basically multicast, position-based routing. In geocast routing, messages are sent from one source node to all other nodes within a ZOR. This routing is used in many VANET applications. For example, in cooperative collision warning systems, a vehicle detects an accident, reports it immediately, and warns nearby vehicles. Vehicles outside the ZOR are not notified in order to avoid unnecessary and hasty reactions [6]. Geocast routing protocols work by defining two zones: the target zone and the forwarding zone.

The forwarding zone includes at least the target zone and a route between the source node and that zone. When an intermediate node receives a message, as in the unicast directional flooding procedure, the message is forwarded provided that the node belongs to the forwarding zone. When it arrives at the target zone, the message receiver broadcasts it to all neighbors only if the node belongs to the target zone and the message has not been received before. Location-based multicast (LBM) [36, 43] is an example of such a routing protocol. GAMER [43], as a geocast protocol, dynamically adjusts the size of the forwarding area depending on the current state of the network.

Mobility-based Routing. In some vehicular applications, such as warning messages for dangerous road surface conditions and unexpected fog banks, suitable broadcast strategies must be defined that are capable of delivering warning messages to the highest number of upstream vehicles in the shortest possible time [44]. To meet these requirements, a mobility-based approach is needed. In this situation, the destination of the message is determined according to the mobility of the vehicles, digital maps, or exchanges of messages. In the optimized adaptive broadcast scheme (OABS) [45], emergency warning messages are transmitted rapidly from an abnormal vehicle to others by adjusting the probability and delay of rebroadcast based on the number of one-hop and two-hop neighbors. Another example of a mobility-based routing protocol is the receive on most stable group path (ROMSGP) protocol [17, 46]. The key idea in this protocol is to group vehicles according to their velocity vectors to ensure that vehicles that belong to the same group are generally moving together and that the routes used by vehicles from the same group have high levels of stability. Then, the protocol determines and sets up the most stable route among the possible routes.

As mentioned earlier, Table 2 lists most of the proposed protocols according to the routing types named in this section. By comparing Table 1 and Table 2, a mismatch between application requirements and proposed protocols can be seen.

5 Conclusions

According to the application requirements, VANET routing protocols are classified as unicast, multicast, and broadcast. Furthermore, regarding selective broadcast behavior of vehicular applications, the classifications can be reduced to unicast and multicast. Each class contains some routing protocols designed for specific requirements. The survey showed that, although all types of protocols are needed, less research has been done on the design of the multicast routing protocol even though most of the applications require this kind of protocol.

In future work, this classification could be improved by considering other parameters. For instance, traffic density (i.e., whether traffic is sparse or dense) may be counted as another factor for refining the classification or, in MIVC, the next hop decision can be considered as sender-oriented (decide before transmitting the message) or receiver-oriented (decide after transmitting the message). Since safety is the most important and frequently used application of VANET, secure routing is another challenging issue that should be studied precisely.

Acknowledgments. This study has been conducted at the computer and network security laboratory, Universiti Kebangsaan Malaysia (UKM). The work was supported by the university through university research grant UKM-AP-ICT-17-2009.

References

1. Peden, M.: World Report on Road Traffic Injury Prevention. World Health Organization (2004)
2. Fatality Analysis Reporting System, <http://www-fars.nhtsa.dot.gov/Main/index.aspx>
3. ICT for smart, safe & clean mobility, <http://www.icarsupport.org/>
4. Lee, K., Lee, U., Gerla, M.: Survey of Routing Protocols in Vehicular Ad Hoc Networks. In: Chapter in Vehicular Ad-hoc Networks: Developments and Challenges (2010)
5. Willke, T.L., Tientrakool, P., Maxemchuk, N.F.: A Survey of Inter-vehicle Communication Protocols and Their Applications. *IEEE Communications Surveys & Tutorials* 11, 3–20 (2009)
6. Li, F., Wang, Y.: Routing in Vehicular Ad Hoc Networks: A Survey. *IEEE Vehicular Technology Magazine* 2, 12–22 (2008)
7. Bensen, J., Manivannan, D.: Unicast Routing Protocols for Vehicular Ad Hoc Networks: A Critical Comparison and Classification. *Pervasive and Mobile Computing* 5, 1–18 (2009)
8. Gongjun, Y., Mitton, N., Li, X.: Reliable Routing in Vehicular Ad hoc Networks. In: The 7th International Workshop on Wireless Ad Hoc and Sensor Networking (2010)
9. Tee, C.A.T.H., Lee, A.C.R.: Survey of Position Based Routing for Inter Vehicle Communication System. In: First International Conference on Distributed Framework and Applications, pp. 174–182 (October 2008)

10. Sichitiu, M.L., Kihl, M.: Inter-Vehicle Communication Systems: A Survey. *IEEE Communications Surveys & Tutorials* 10, 88–105 (2008)
11. Khaled, Y., Tsukada, M., Santa, J., Choi, J., Ernst, T.: A Usage Oriented Analysis of Vehicular Networks: from Technologies to Applications. *Journal of Communications* 4, 357–368 (2009)
12. Toor, Y., Muhlethaler, P., Laouiti, A.: Vehicle Ad Hoc Networks: Applications and Related Technical Issues. *IEEE Communications Surveys & Tutorials* 10, 74–88 (2008)
13. Hartenstein, H., Laberteaux, K.P.: A Tutorial Survey on Vehicular Ad Hoc Networks. *IEEE Communications Magazine* 46, 164–171 (2008)
14. Yousefi, S., Mousavi, M.S., Fathy, M.: Vehicular Ad Hoc Networks (VANETs): Challenges and Perspectives. In: *Proceedings of the 6th International Conference on ITS Telecommunications*, pp. 761–766 (2006)
15. Caveney, D.: Cooperative Vehicular Safety Applications. *IEEE Control Systems Magazine* 30, 38–53 (2010)
16. Wolf, M.: *Security Engineering for Vehicular IT Systems*. Vieweg and Teubner, Germany (2009)
17. Huang, C., Chang, Y.: *Telematics Communication Technologies and Vehicular Networks: Wireless Architectures and Applications*. Information Science Reference-Imprint of: IGI Publishing Hershey, PA (2009)
18. Olariu, S., Weigle, M.: *Vehicular Networks: From Theory to Practice*. Chapman & Hall/CRC, Boca Raton (2009)
19. Plossl, K., Nowey, T., Mletzko, C.: Towards a Security Architecture for Vehicular Ad Hoc Networks. In: *The First International IEEE Conference on Availability, Reliability and Security*, p. 8 (2006)
20. Moustafa, H., Zhang, Y.: *Vehicular Networks: Techniques, Standards, and Applications*. Auerbach Publications Boston, MA, USA (2009)
21. By, E., Guo, H., Cover, H., Access, P.: *Automotive Informatics and Communicative Systems: Principles in Vehicular Networks and Data Exchange*. Information Science Reference, Hershey, New York (2009)
22. Popescu-Zeletin, R., Radusch, I., Rigani, M.: *Vehicular-2-X Communication: State-of-the-Art and Research in Mobile Vehicular Ad Hoc Networks*. Springer, Heidelberg (2009)
23. Tonguz, O.K., Wisitpongphan, N., Parikh, J.S., Fan, B., Mudalige, P., Sadekar, V.K.: On the Broadcast Storm Problem in Ad hoc Wireless Networks. In: *3rd International Conference on Broadband Communications, Networks and Systems*, pp. 1–11 (2006)
24. Chen, R., Wen-Long, J., Regan, A.: Broadcasting Safety Information in Vehicular Networks: Issues and Approaches. *IEEE Network* 24, 20–25 (2010)
25. Abolhasan, M., Wysocki, T., Dutkiewicz, E.: A Review of Routing Protocols for Mobile Ad Hoc Networks. *Ad Hoc Networks* 2, 1–22 (2004)
26. Namboodiri, V., Lixin, G.: Prediction-Based Routing for Vehicular Ad Hoc Networks. *IEEE Transactions on Vehicular Technology* 56, 2332–2345 (2007)
27. Benzaid, M., Minet, P., Al Agha, K.: Integrating Fast Mobility in the OLSR Routing Protocol. In: *4th International Workshop on Mobile and Wireless Communications Network*, pp. 217–221 (2002)
28. Fukuhara, T., Warabino, T., Ohseki, T., Saito, K., Sugiyama, K., Nishida, T., Eguchi, K.: Broadcast Methods for Inter-vehicle Communications System. In: *IEEE Wireless Communications and Networking Conference*, pp. 2252–2257 (2005)
29. Yang, Q., Lim, A., Li, S., Fang, J., Agrawal, P.: Acar: Adaptive Connectivity Aware Routing for Vehicular Ad Hoc Networks in City Scenarios. *Mobile Networks and Applications* 15, 36–60 (2010)

30. Granelli, F., Boato, G., Kliazovich, D.: MORA: A Movement-based Routing Algorithm for Vehicle Ad Hoc Networks. In: 1st IEEE Workshop AutoNet (2006)
31. Mo, Z., Zhu, H., Makki, K., Pissinou, N.: MURU: A Multi-Hop Routing Protocol for Urban Vehicular Ad Hoc Networks. In: Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services, pp. 1–8 (2006)
32. Burgess, J., Gallagher, B., Jensen, D., Levine, B.N.: MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In: 25th IEEE International Conference on Computer Communications, pp. 1–11 (2006)
33. Prasanth, K., Duraiswamy, K., Jayasudha, K., Chandrasekar, C.: Minimizing End-to-end Delay in Vehicular Ad Hoc Network Using Edge Node Based Greedy Routing. In: First International Conference on Advanced Computing, pp. 135–140 (2009)
34. Jerbi, M., Senouci, S.M., Rasheed, T., Ghamri-Doudane, Y.: Towards Efficient Geographic Routing in Urban Vehicular Networks. *IEEE Transactions on Vehicular Technology* 58, 5048–5059 (2009)
35. Joshi, H.P.: Distributed Robust Geocast: A Multicast Protocol for Inter-vehicle Communication. Dept. of Computer Networking and Electrical Engineering, Master's Thesis. NCSU (2007)
36. Wai, C., Guha, R.K., Taek Jin, K., Lee, J., Hsu, I.Y.: A Survey and Challenges in Routing and Data Dissemination in Vehicular Ad-Hoc Networks. In: IEEE International Conference on Vehicular Electronics and Safety, pp. 328–333 (2008)
37. Oh, S., Kang, J., Gruteser, M.: Location-Based Flooding Techniques for Vehicular Emergency Messaging. In: Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services, pp. 1–9 (2006)
38. Harsch, C., Festag, A., Papadimitratos, P.: Secure Position-Based Routing for VANETs. In: 66th IEEE Vehicular Technology Conference, pp. 26–30 (2007)
39. Sun, W., Yamaguchi, H., Yukimasa, K., Kusumoto, S.: GVGrid: A QoS Routing Protocol for Vehicular Ad Hoc Networks. In: 14th IEEE International Workshop on Quality of Service, pp. 130–139 (2006)
40. Seet, B.-C., Liu, G., Lee, F.B.S., Foh, C.-H., Wong, K.-J., Lee, K.-K.: A-STAR: A Mobile Ad Hoc Routing Strategy for Metropolis Vehicular Communications. In: Mitrou, N.M., Kontovasilis, K., Rouskas, G.N., Iliadis, I., Merakos, L. (eds.) NETWORKING 2004. LNCS, vol. 3042, pp. 989–999. Springer, Heidelberg (2004)
41. Jing, Z., Guohong, C.: VADD: Vehicle-Assisted Data Delivery in Vehicular Ad Hoc Networks. *IEEE Transactions on Vehicular Technology* 57, 1910–1922 (2008)
42. Wu, H., Fujimoto, R., Guensler, R., Hunter, M.: MDDV: A Mobility-centric Data Dissemination Algorithm for Vehicular Networks. In: 1st ACM International Workshop on Vehicular Ad Hoc Networks, pp. 47–56 (2004)
43. Maihofer, C.: A Survey of Geocast Routing Protocols. *IEEE Communications Surveys & Tutorials* 6, 32–42 (2004)
44. Fasolo, E., Zanella, A., Zorzi, M.: An Effective Broadcast Scheme for Alert Message Propagation in Vehicular Ad hoc Networks. In: IEEE International Conference on Communications, pp. 3960–3965 (2006)
45. Alshaer, H., Horlait, E.: An Optimized Adaptive Broadcast Scheme for Inter-vehicle Communication. In: 61st IEEE Vehicular Technology Conference, vol. 2845, pp. 2840–2844 (2005)
46. Taleb, T., Sakhaee, E., Jamalipour, A., Hashimoto, K., Kato, N., Nemoto, Y.: A Stable Routing Protocol to Support ITS Services in VANET Networks. *IEEE Transactions on Vehicular Technology* 56, 3337–3347 (2007)

Comparative Study on the Performance of TFRC over AODV and DSDV Routing Protocols

Khuzairi Mohd Zaini¹, Adib M. Monzer Habbal¹, Fazli Azzali¹,
and Mohamad Rizal Abdul Rejab²

¹ UUM College of Arts and Sciences, Universiti Utara Malaysia,
06010 Sintok, Kedah Darulaman, Malaysia
{khuzairi, adib, fazli}@uum.edu.my

² School of Communications and Computer Engineering,
Universiti Malaysia Perlis Malaysia
m.rizal@unimap.edu.my

Abstract. Nowadays, many researches in Mobile ad hoc network (MANET) focus mainly on the impact of MANET routing protocol to TCP variants, but none have been done on TFRC. In this paper, we present a comprehensive simulation study on the behavior of TCP-friendly Rate Control (TFRC) protocol over Ad-Hoc On Demand Vector (AODV) routing protocol and Destination-Sequenced Distance Vector (DSDV). The main objective is to measure the TFRC performance in terms of throughput, jitter and delay. In addition, this research also identifies whether or not MANET routing protocols have impact on TFRC. We conduct the experiment by sending multimedia streaming traffic carried by TFRC over AODV and DSDV respectively. Random-Waypoint mobility model is used for both experiments. Results obtained shows that TFRC has better throughput over DSDV. As for delay and jitter, TFRC over AODV has smoother results.

Keywords: Mobile Ad-Hoc Network; AODV; DSR; TFRC; Performance Evaluation.

1 Introduction

Wireless mobile network can be broadly categorized into infrastructure based with central administration and distributed coordination function without central administration [1]. Mobile wireless ad hoc network (MANET) falls under the latter category which is a self-organized and dynamically reconfigurable wireless network of mobile nodes [2]. Nodes in MANET can act as end hosts or intermediary nodes. In MANET, all nodes can instantly establish communication and topology can potentially change due to mobility and disappearance of nodes.

Similar with other nodes in different network architectures, MANET's nodes also implement TCP/IP protocol suit for their communications. A part from physical and data link layer, MANET heavily depends on the routing and transport protocols. MANET is different from wired network in terms of the characteristics such as

half-duplex links, channel noise, mobility and hidden terminal problem, route change and disconnection [3]. Therefore, these distinguish the routing protocols used in MANET.

Nodes in MANET may utilize TCP or UDP as their transport protocols depending on types of applications. However, TCP has undergone several enhancements to make it suitable working in the wireless environment. A comprehensive survey of TCP enhancement in wireless network can be found in the article by [4] and [28]. Unlike TCP, which adjust the sending rate according to the traffic, UDP is considered as a greedy protocol. Thus, some still consider TCP to carry their multimedia traffics [5] in order to maintain the stability of the Internet. However, multimedia traffics carried over TCP suffer from low quality of service since TCP does not reply smoothly in the dynamic changing of network especially in wireless environment.

In response to the problems of TCP and UDP, a new transport protocol was proposed, namely TCP-Friendly Rate Control protocol [6]. The prime objectives of TFRC are to be friendly to TCP flows and at the same time maintain the smoothness of the changing rate to avoid severe performance degradation. TFRC is envisioned to be the choice of transport protocol for inelastic applications.

Most of the previous researchers studied and evaluated the performance of transport protocols in isolation from MANET's routing protocols. For example, just to name a few, [7, 8, 9, 10, 11, 12]. Recently, researchers have started to take a look on the interaction between transport protocols and other networking layers, e.g. [13, 14, 15, 16]. Although many efforts have been established to study the relationship of TCP and different MANET routing protocols, but, to the best of our knowledge, none have been done on TFRC.

This paper is organized as follows. Section 2 briefly describes MANET routing protocols, TFRC and related work. Section 3 discusses the simulation topology and its parameter. Performance metrics also discusses in this section. Section 4 discusses simulation results of the experiment and Section 5 concludes this paper.

2 Background and Related Work

E. M. Royer and T. Chai-Keong [1] classified MANETs routing protocols into Table-driven and Source-initiated, while [17] classified MANET routing protocols into three categories, namely global/proactive, on-demand/reactive and hybrid. In proactive routing protocols, the routes to all the destination (or parts of the network) are determined at the start up, and maintained by using a periodic route update process. Reactive protocols attempt to minimize the overhead of proactive routing protocols by maintaining and updating for active routes only. In reactive protocols, routes are determined when they are required by the source using a route discovery process. Hybrid routing protocols combine the basic properties of the first two classes of protocols into one. Each group has a number of different routing strategies, which employ a flat or a hierarchical routing structure.

Based on our literature review [22,23,24], it is found that the performance of AODV protocol is better than DSR and DSDV protocols in terms of end-to-end delay

and routing overhead metrics because AODV is an on demand protocol where it builds routes only as desired by source nodes. However, in terms of packet fraction ratio, DSDV has better result. Table 1 summarizes the performance comparison between AODV, DSR and DSDV in terms of packet delivery ratio, end-to-end delay and routing overhead.

Table 1. Performance Comparison of AODV, DSDV and DSR

Routing Protocols	Packet Delivery Fraction	End-to-end Delay	Routing Overhead
AODV	High	Low	Low
DSDV	Low	High	High
DSR	High	Low	Low

2.1 Ad-Hoc On Demand Distance Vector (AODV)

Ad hoc On Demand Distance Vector (AODV) is capable of unicast and multicast routing [20]. It is an on demand algorithm, meaning that the routes between nodes are only created as desired by source nodes. Firstly, Hello messages will be used to detect and links to the neighbors. Each active node periodically broadcasts a Hello message to all its neighbors. Because nodes periodically send Hello messages, if a node fails to receive several Hello messages from a neighbor, a link break is detected.

Secondly, when a source has data to transmit to an unknown route, it broadcasts a Route Request (RREQ) for that destination. At each intermediate node, when a RREQ is received a route to the source is created. If the receiving node has not received this RREQ before, and the node is not the destination and does not have a current route to the destination, it rebroadcasts the RREQ. If the receiving node is the destination or has a current route to the destination, it generates a Route Reply (RREP). The RREP is unicast in a hop-by hop fashion to the source. As the RREP propagates, each intermediate node creates a route to the destination. When the source receives the RREP, it records the route to the destination and can begin sending data. If multiple RREPs are received by the source, the route with the shortest hop count is chosen.

As data flows from the source to the destination, each node along the route updates the timers associated with the routes to destination, maintaining the routes in the routing table. If a route is not used for some period of time, a node cannot be sure whether the route is still valid; consequently, the node removes the route from its routing table.

If data is flowing and a link break is detected, a Route Error (RERR) is sent to the source of the data in a hop-by hop fashion. As the RERR propagates towards the source, each intermediate node invalidates routes to any unreachable destinations. When the source of the data receives the RERR, it invalidates the route and reinitiates route discovery if necessary.

2.2 Destination-Sequenced Distance-Vector (DSDV)

DSDV is a proactive routing protocol. It is a routing protocol based on the Bellman-Ford algorithm developed by C. Perkins and P. Bhagwat in 1994 [21]. The Algorithm can be used to solve the routing looping problem that happened in the networks. Each node periodically broadcasts routing updates. A sequence number is assigned for each entry. A route with higher sequence number is more favorable. Thus, two routes with same sequence number the one with lowest hops is more favorable. If a node detects that route to a destination has broken, then its hop number is set to infinity and its sequence number increased but assigned an odd number. Even numbers correspond to sequence number of connected paths. DSDV is not suitable for highly dynamic networks, whenever the topology of the network changes.

2.3 TCP-Friendly Rate Control (TFRC)

Unlike TCP which uses window-based congestion control, TFRC uses equation-based mechanism. It has been designed to adapt the sending rate of a flow in a smooth manner, while trying to fairly share the available bandwidth with competing TCP flows [25]. A TFRC sender adjusts its rate as a function of the measured rate of loss events, where a loss event consists of one or more packets dropped within a single round-trip time.

3 Simulation and Performance Metrics

In this section, we present the network scenarios and parameters used in this simulation study as well as the metrics for performance evaluation.

3.1 Simulation Scenarios and Parameters

The simulation parameters are summarized in Table 2. We conducted two sets of experiments: without and with background traffic. The background traffics have five UDP connections. In all scenarios, random waypoint mobility model is used with the

Table 2. Simulation Parameters

Parameters	Setting
Routing Protocols	AODV, DSDV
MAC Protocol	IEEE 802.11
Simulation Time	500 seconds
Simulation Area	1000m X 1000m
Mobility Model	Random Waypoint
Number of Nodes	50
Packet Size	1000 bytes
Packet Sent rate	0.01Mbps
NS-2 version	2.34

speed of 5,10, 15 and 20 m/s. As for the pause time, we use 0, 10 and 20 second. The transmitted traffic is Constant Bit Rate (CBR) to represent the multimedia streaming types of traffic. We first carry CBR traffic by using TFRC over AODV and followed by DSDV. A random pair of sender and receiver using TFRC was selected. We have averaged several different mobility scenarios to obtain each result. We choose NS-2 as the simulation tool where 44% of the MANET research communities use it [26].

3.2 Performance Metrics

The performance metrics use to measure in this experiments are:

Throughput

The TFRC throughput equation is based on the TCP Reno equation from [18]:

$$X = \frac{s}{R\sqrt{\frac{2bp}{3}} + t_{RTO} \left(3\sqrt{\frac{3bp}{8}} \right) p(1+32p^2)} \quad (1)$$

Where, X is the transmission rate in bytes/second, b is the number of packets acknowledged by a single TCP ACK, R is the RTT in seconds and s is the packet size in bytes. t is the TCP retransmission timeout value in seconds, which is set to $4R$ in practice. p is the loss rate, which is between 0 and 1.0, of the number of loss events as a fraction of the number of packets transmitted.

Packet Delay

Packet delay is a combination of delays caused by processing, transmission, and queuing delays in routers, end-system processing delays, and propagation delays in the links. Unlike in wired network, queuing delay rarely happen in MANET. Therefore, the end-to-end delay is measured using the following formula [27]:

$$d_{\text{end-end}} = N (d_{\text{proc}} + d_{\text{trans}} + d_{\text{prop}}) \quad (2)$$

Where, d_{prop} = propagation on each link, d_{proc} = processing delay at each router and at the source hosts. $d_{\text{trans}} = L/R$, where L = packet size and R = link rate of transmission (bits/second).

Jitter

Delay variation occurs when the time taken by an IP datagram to travel from source to destination varies from one datagram to the next datagram because of variable delay [19]. This variation is called delay variation or jitter. A high jitter can have a severe impact on the performance of multimedia applications.

4 Experiments Results

In this section, we discuss the results obtained from the simulation experiments.

4.1 Scenario without Background Traffic

We observed that the average throughput of TFRC over DSDV is more than AODV in all speeds value. As shown in Fig. 1, the graph pattern of TFRC over AODV and DSDV are similar, where the throughput decreases when the node speed move from 5 to 10, increases as the speed goes to 15 and decreases at the speed of 20. More importantly, as the speed goes higher i.e. speed 15 and 20 the average differences throughput of TFRC via DSDV is 64% and 70% higher than AODV.

Although TFRC over DSDV has better throughput, but in terms of jitter and delay, TFRC over AODV produces better result. As shown in Fig. 2 and Fig. 3, the delay and jitter increase tremendously when TFRC use DSDV as the speed change from 5 to 10.

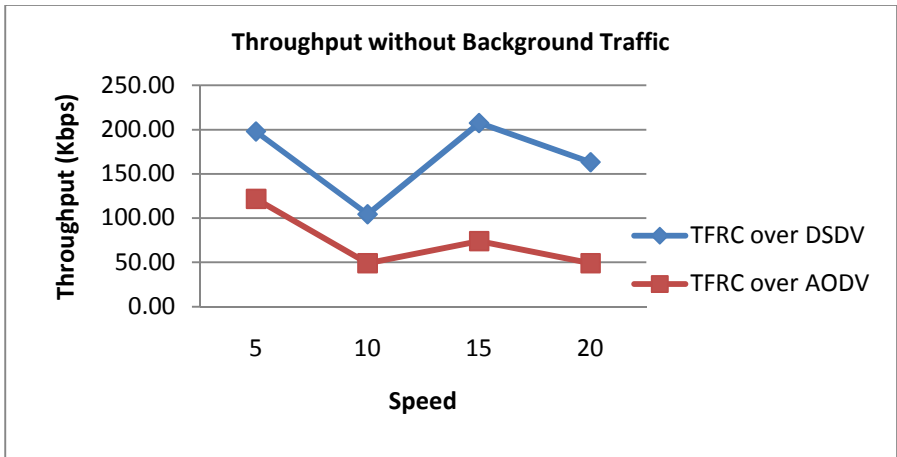


Fig. 1. Throughput without background Traffic

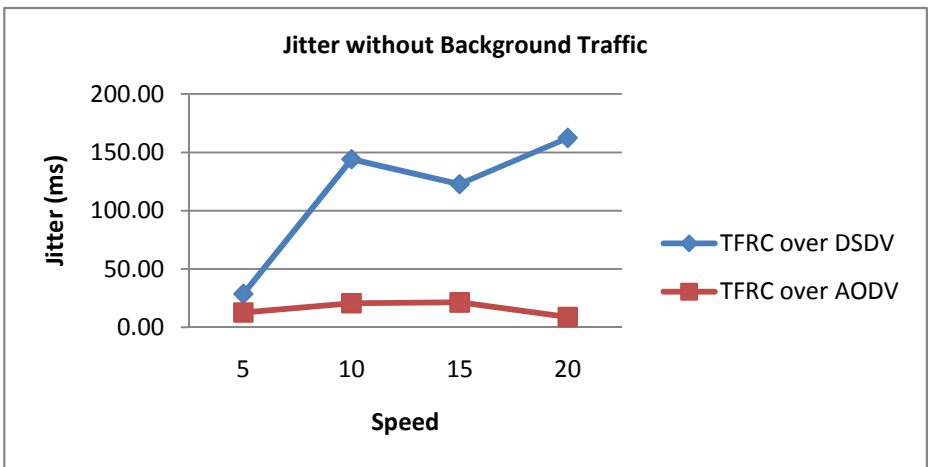


Fig. 2. Jitter without background Traffic

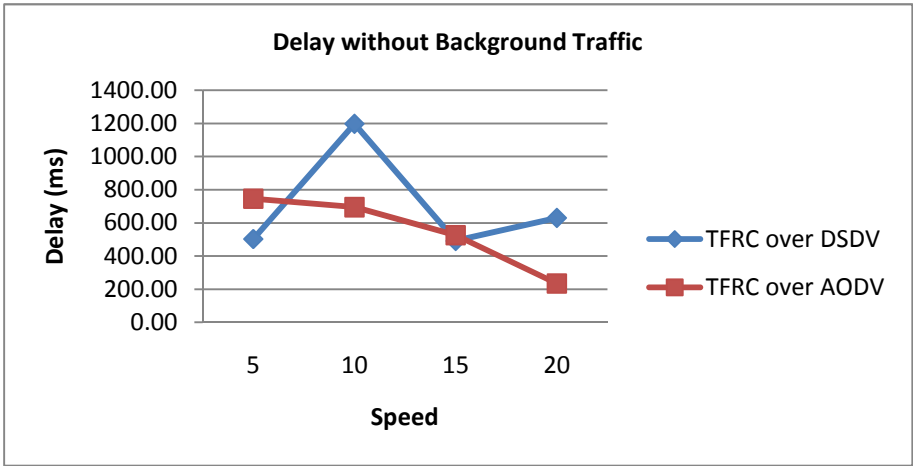


Fig. 3. Delay without background Traffic

4.2 Scenario with Background Traffic

The average throughput of TFRC via DSDV decreases as the speed increases. At speed 5, the average throughput of TFRC over DSDV is 100% more than the throughput of TFRC via AODV. Fig. 4 shows that when the speed changes from 5 to 10, the throughput of TFRC over DSDV drops significantly which is almost 60%.

Similar with the previous experiment, TFRC over AODV gives better result for jitter and delay, which can be found in Fig. 5 and Fig. 6.

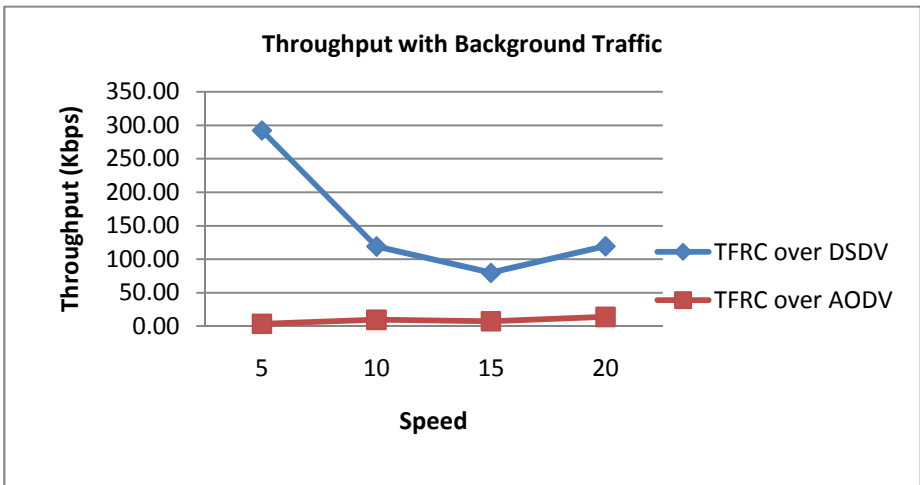


Fig. 4. Throughput with background Traffic

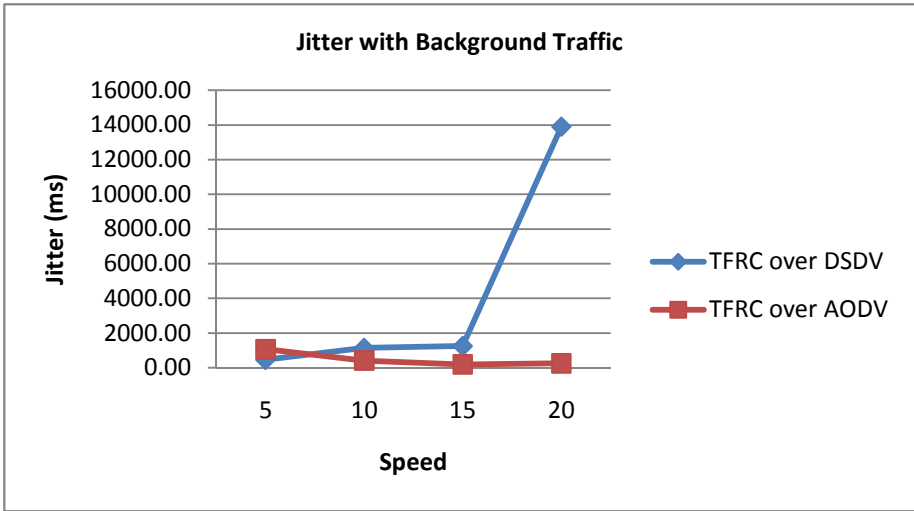


Fig. 5. Jitter with background Traffic

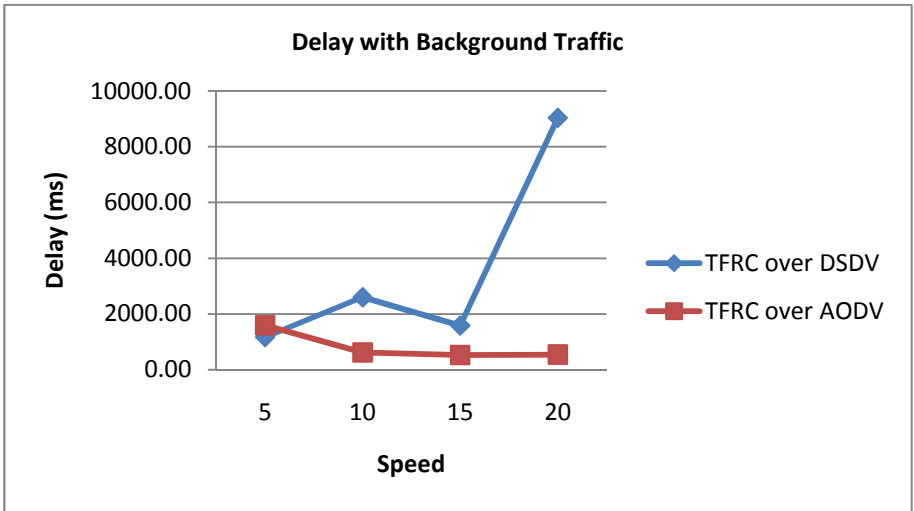


Fig. 6. Delay with background Traffic

4.3 Discussion

Irrespective of with or without background traffic, TFRC over DSDV always have higher throughput due to the facts that DSDV always have alternative routes in advance. However, the jitter and delay of TFRC over DSDV are very poor because in this dynamic environment, DSDV updates its routing table frequently, and whenever a new packet arrives it will be forwarded through the newly discovered routes to the same destination. On the other hand, AODV continuously use the same route till route failure occurs which results in low throughput, but smoother jitter and delay.

5 Conclusion

The main objective of our study is to identify whether or not MANET routing protocols have impact on the performance of TFRC. As a comparison, we run TFRC over AODV and DSDV routing protocols. Based on the comprehensive simulations conducted, we found that different routing protocols have given different TFRC performance in terms of throughput, delay and jitter. TFRC over DSDV gives better throughput but TFRC over AODV produces better delay and jitter. In summary, TFRC over DSDV is more suitable to carry elastic applications, while TFRC over AODV is more preferable for multimedia applications.

In the future, we will investigate the smoothness of TFRC and TCP New-Reno over different MANET routing protocols.

Acknowledgement

This work is fully supported by Universiti Utara Malaysia under the LEADS research grant scheme.

References

1. Royer, E.M., Chai-Keong, T.: A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks. *IEEE Personal Communications* 6, 46–55 (1999)
2. Mahmud, S.A., Khan, S., Khan, S., Al-Raweshidy, H.: A Comparison of MANETs and WMNs: Commercial Feasibility of Community Wireless Networks and MANETs. In: *Proceedings of the 1st International Conference on Access Networks*, Athens, Greece (2006)
3. Nahm, K., Helmy, A., Kuo, C.-C.J.: TCP over Multihop 802.11 Networks: Issues and Performance Enhancement. In: *Proceedings of the 6th ACM international Symposium on Mobile Ad Hoc Networking and Computing*, Urbana-Champaign, IL, USA, pp. 277–287 (2005)
4. Al Hanbali, A., Altman, E., Nain, P.: A survey of TCP over ad hoc networks. *IEEE Communications Surveys & Tutorials* 7, 22–36 (2005)
5. Wang, B., Kurose, J., Shenoy, P., Towsley, D.: Multimedia streaming via TCP: An analytic performance study. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)* 4, 1–22 (2008)
6. Handley, M., Padhye, J., Floyd, S., Widmer, J.: TCP friendly rate control (TFRC): Protocol specification (2001)
7. Dyer, T.D., Boppana, R.V.: A Comparison of TCP Performance over Three Routing Protocols for Mobile Ad Hoc Networks. In: *MobiHoc 2001: Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing*, Long Beach, CA, UAS, pp. 56–66 (2001)
8. Luo, J., Peng, H., Tang, H.: Performance study of TCP VenO in MANET environment. *Jisuanji Gongcheng/ Computer Engineering* 35, 104–106 (2009)
9. Qamar, S., Manoj, K.: Impact of Random Loss on TCP Performance in Mobile Ad hoc Networks (IEEE 802.11), A Simulation-Based Analysis. Arxiv preprint arXiv:1002.2403 (2010)
10. Xiao-qin, Z., Liang, L.: Study on TCP Performance Improvement over MANET. *Communications*, p. 7 (2009)

11. Al Hanbali, A., Kherani, A., Groenovel, R., Nain, P., Altman, E.: Impact of Mobility on the Performance of Message Relaying in Ad Hoc Networks. Technical Report RR-5480, INRIA, Sophia-Antipolis (2005)
12. Kumar, A., Jacob, L., Ananda, A.: SCTP vs TCP: Performance comparison in MANETs. In: Proceedings 29th Annual IEEE International Conference on Local Computer Networks (2004)
13. Ahuja, A., Agarwal, S., Singh, J., Shorey, R.: Performance of TCP over Different Routing Protocols in Mobile ad-hoc Networks. In: IEEE 51st Presented at Vehicular Technology Conference Proceedings, Tokyo (2002)
14. Kim, D., Bae, H., Song, J., Cano, J.: Analysis of the interaction between TCP variants and routing protocols in MANETs. In: Proc. International IEEE Conf. Parallel Processing Workshops, ICPPW (2005)
15. Seddik-Ghaleb, A., Ghamri-Doudane, Y., Senouci, S.: Effect of ad hoc routing protocols on TCP performance within MANETs. In: 2006 3rd Annual IEEE Communications Society on Presented at Sensor and Ad Hoc Communications and Networks, SECON 2006 (2007)
16. Abolhasan, M., Wysocki, T., Dutkiewicz, E.: A review of routing protocols for mobile ad hoc networks. *Ad Hoc Networks* 2, 1–22 (2004)
17. Noorani, R., Ansari, A., Khowaja, K., Laghari, S., Shah, A.: Analysis of MANET Routing Protocols under TCP Vegas with Mobility Consideration. *Wireless Networks, Information Processing and Systems*, 227–234 (2009)
18. Padhye, J., Firoiu, V., Towsley, D., Kurose, J.: Modeling TCP Throughput: A Simple Model and its Empirical Validation. In: Proceedings of the ACM SIGCOMM 1998 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, Vancouver, British Columbia, Canada, pp. 303–314 (1998)
19. Hassan, M., Jain, R.: *High Performance TCP/IP Networking: Concepts, Issues, and Solutions*. Pearson Prentice Hall, Upper Saddle River (2004)
20. Perkins, C.E., Royer, E.M.: Ad Hoc On- Demand Distance Vector Routing. In: Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, pp. 90–100 (1999)
21. Perkins, C.E., Bhagwat, P.: Highly Dynamic Destination-Sequenced Distance Vector Routing (DSDV) for Mobile Computers. In: Proceedings of ACM SIGCOMM 1994, London, pp. 234–244 (1994)
22. Abd Rahman, A.H., Zukarnain, Z.A.: Performance Comparison of AODV, DSDV and I-DSDV Routing Protocols in Mobile Ad-hoc Networks. *European Journal of Scientific Research* 31(4), 566–576 (2009), ISSN 1450-216X
23. Hassan, Y.K., Abd El Aziz, M.H., Abd El-Radi, A.S.: Performance Evaluation of Mobility Speed over MANET Routing Protocols. *International Journal of Network Security* 11(3), 128–138 (2010)
24. Yadav, N.S.: The Effects of Speed on the Performance of Routing Protocols in Mobile Ad-hoc Networks. *International Journal of Electrical and Information Engineering*, 79–84 (2007)
25. Floyd, S., Handley, M., Padhye, J., Widmer, J.: Equation-based congestion control for unicast applications. In: ACM SIGCOMM 2000, Stockholm, pp. 43–56 (August 2000)
26. Kurkowski, S., Michael, T.C., Colagrosso, M.: MANET Simulation Studies: The Incredibles. *Mobile Computing and Communications Review, Networking and Mobile Computing* 9(4), 1–12 (2005)
27. Kurose, J.F., Ross, K.W.: *Computer Networking: A Top-Down Approach*, 5th edn. Pearson Addison-Wesley, Boston (2010)
28. Habbal, A.M.M., Hassan, S.: Loss Detection and Recovery Techniques for TCP in Mobile Ad Hoc Network. In: proceedings of the 2nd International Conference on Network Applications Protocols and Services (NETAPPS), Alor Setar, Malaysia, pp. 48–54 (2010)

An Enhanced Route Discovery Mechanism for AODV Routing Protocol

Kamarularifin Abd. Jalil¹, Zaid Ahmad², and Jamalul-Lail Ab Manan²

¹ Faculty of Computer and Mathematical Sciences,
Universiti Teknologi MARA Malaysia, Shah Alam, Selangor, Malaysia
kamarul@tmsk.uitm.edu.my

² Advanced Information Security, MIMOS Berhad
Technology Park Malaysia, Kuala Lumpur, Malaysia
{zaid.ahmad,jamalul.lail}@mimos.my

Abstract. Due to the unique characteristic of Mobile Ad hoc Network (MANET) and lack of security in its routing protocol, MANET is vulnerable to various attacks such as black hole. In this paper we study a black hole attack on one of ad hoc routing protocol called AODV (Ad hoc On Demand Vector). There have been many works done to solve this problem but most of them introduced extra overheads. In this paper we proposed a novel method to address this limitation called ERDA (Enhance Route Discovery for AODV) by improving the route discovery mechanism in the AODV protocol. The first part of this method is to secure the routing table update by introducing new parameter called `rt_upd` in `recvReply()` algorithm of AODV. The second part is to analyze AODV Receive Reply messages stored in a table called `rrep_tab` to isolate malicious nodes by maintaining those nodes in a list called `mali_list`. ERDA provides secure and low latency of route discovery as compared to previous methods. One of our future works is to perform a simulation to determine ERDA performance against other proposed methods in protecting MANET from black hole attacks.

Keywords: black hole attack; ad hoc on-demand distance vector; mobile ad hoc networks; route discovery; ERDA; MANET; receive reply message; AODV.

1 Introduction

A mobile ad hoc network (MANET) is a group of mobile devices connected by wireless link, mostly are in temporary manner. It is formed in situation where creating the network infrastructure would be impossible or prohibited by certain reasons. It does not require fix infrastructure or centralised management like access point or based-station. The networks' nodes can be static but most of the time they are mobile and dynamic. Mobile devices or nodes in the network are free to enter or leave the network. As a result, network topology will change frequently. In MANET, mobile nodes which are in the same communication range can communicate directly amongst them. On the other hand, if the communication is out of range, they still able to communicate but require cooperation from other nodes to relay their messages by

using the multi-hop network. As such, every node in MANET has two roles, i.e., as a host and as a router. In a multi-hop network, routing protocol will play very important role as an enabler for end-to-end communication. The Ad hoc On-demand Distance Vector (AODV) protocol [1] is an example of an ad hoc routing protocol [2] available today.

Due to the unique characteristics of MANET, the network is very attractive to users. Unfortunately, the MANET is also vulnerable to attack like any other wireless networks. This is due to the fact that wireless networks use the ether to propagate information which is also accessible by the malicious nodes. Moreover, it is hard to differentiate between normal and abnormal activities in a mobile background because the lack of security features in the existing routing protocol i.e. the AODV. It is easy for compromised or malicious node to inject false routing information to the network with the intention to deny the service or to eavesdrop messages. Thus, the security in a routing protocol such as the AODV is very crucial and the researchers all around the world are having hard time developing a secure and efficient routing protocol for this network. Black hole is one of many attacks that take place in MANET and is considered as one of the most common attacks made against the AODV routing protocol. The black hole attack involves malicious node pretending to have the shortest and freshest route to the destination by constructing false sequence number [3] in control messages.

AODV protocol was created without any security considerations [4]. Thus, no protection mechanism was built to detect the existence of malicious attack. In the AODV, maintaining a fresh route to ensure safe path to destination is very vital due to the rapid change of the network topology. Thus as mention earlier, the destination sequence number and the number of hop are vital attributes to determine the freshness of the route. In this case, the most trusted sequence number will be the one coming from the actual destination node. In Section 2, the detail information about the routing update in AODV discovery process will be explained. However, it was observed that the current mechanism used is not efficient enough since it is based on the largest sequence number and less hops count. Such mechanism would open up the opportunity to attackers to manipulate it. This could be seen in a possible scenario of a black hole attack, where a malicious node will be the first to response to the Route Request (RREQ) message with the fake highest destination sequence number. The manipulation done by the malicious node will deny the genuine Route Reply (RREP) message from other nodes especially the reply message coming from the actual destination node.

In this paper, we study various methods proposed in the previous works to overcome the black hole attack in the AODV-based MANET. Most of the works done to solve this problem have some limitations and are costly i.e. most of them have extra overhead in the processing time during route discovery phase. We have devised a novel method to secure the network from the black hole attack by enhancing the existing AODV route discovery process. We proposed an improved algorithm of the `recvReply()` function in the AODV routing protocol which we called as ERDA (Enhanced AODV). As described in Section 4, the proposed algorithm will has minimum modification to the conventional AODV protocol and less delay.

This paper is organised as follows. Section 2 provides an overview of the AODV route discovery process and a description of a black hole attack. Section 3 discusses

about related works. Section 4 presents the proposed ERDA, a new method to detect and prevent the black hole attack. And lastly, the plan for the future work is concluded in Section 5.

2 Ad Hoc On-Demand Distance Vector

AODV is categorised as a dynamic reactive routing protocol [5]. In a reactive routing protocol, route will be established based on the demand (upon request by source node). The process to discover routing path to the destination node is illustrated in Figure 1. In AODV route discovery, there are two important control messages namely Route Request (RREQ) and Route Reply (RREP). Both control messages carry an important attribute called destination sequence number and has the incremental value to determine the freshness of a particular route.

2.1 Route Discovery Process

In this illustration, the source node S will broadcast control packets, RREQ message to its neighbours A, B and C in order to find the best possible path to the destination node D. Upon receiving the RREQ message, the received node either:

- a) replies to the source node with a RREP message if the received node is the destination node or an intermediate node with a 'fresh enough' route information to the destination, or
- b) updates the routing table entry which will be used in the reverse path and the rebroadcasting of the RREQ message until the destination node or intermediate node with 'fresh enough route' is reached .

An intermediate node is believed to have a 'fresh enough routes' to the destination node if the destination sequence number in its routing table is greater than or equal (with less hop count) to the destination sequence number in the RREQ message.

As mentioned in section 2.1.a above, upon receiving the RREQ message from node A, the destination node D will reply with the RREP message to node S by forwarding the message to node A. In turn, node A will forward the message to the source node S. Once the source node S received the RREP message, it will process the message by calling the AODV `recvReply()` function. This function will update the route entry for destination D if either one of this condition is met.

- a) The destination sequence number in the routing table is less than the destination sequence in the RREP message or
- b) The destination sequence number in the routing table is equal with the destination sequence number in the RREQ message but the hop count is less than the one in the routing table.

In case where node S received multiple RREP messages, this function will select the RREP message with the highest destination sequence number value. The detail mechanism of `recvReply()` function is explained in the Pseudo Code given in Figure 2.

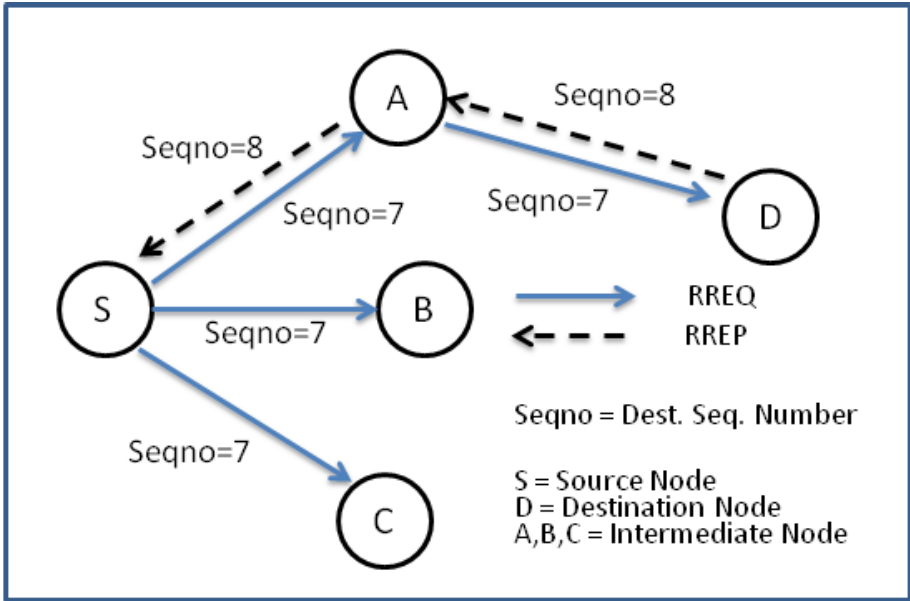


Fig. 1. AODV route discovery process

```

AODV
1  RecvReply(Packet P) {
2  if (P.dst no entry in Routing Table RT) {
3    Add entry of P.dst to RT
4  }
5  select dst_seqno from RT
6  if (P.dst_seqno > RT.dst_seqno or
7    P.dst_seqno = RT.dst_seqno and P.hops < RT.hops) {
8    update RT entry with P
9    send data packets to the route in RT
10 }
11 else if (routing is UP for P) {
12   forward packet P
13   else discards P
14 }
15 }
    
```

Fig. 2. Pseudo code for recvreply() function in AODV

2.2 Black Hole Attack

A black hole attack is a kind of denial of service attack [6] where a malicious node can falsely claiming it has a ‘fresh enough route’ information to the destination. The modus operandi of a black hole attack in the AODV is by attacking the control message sent during the route discovery process whereby a forged RREP message is sent out to catch the attention of other nodes. Apparently, the malicious node will claim that it has the ‘fresh enough route’ information to the destination. If the other nodes fall into this trap, they will send their data packets through the malicious node. The diagram in Figure 3 demonstrates how the malicious node M pretends to be the node with a ‘fresh enough route’ to the destination node D. Upon receiving the RREQ message from node C, node M will generate the RREP message and send it immediately to source node S. The message will contain the faked destination sequence number. There will be more than one RREP messages replied and in order to be favoured against others, the destination sequence number from node M normally higher. In addition, to ensure that it is ahead from the rest of the nodes in sending out the RREP messages, the malicious node will ignore its routing table checking. The source node S will update its routing table, by assuming that the first RREP received is the shortest and freshest path to destination node D. As a result, node S will take the node C as the next hop (malicious path) to send its data to the destination node D. Node C will then forward that data packet to node M. Upon receiving the data packet, node M either will keep or drop the packet without forwarding to the destination node D as if the packet is swallowed by a black hole as the attack name implies.

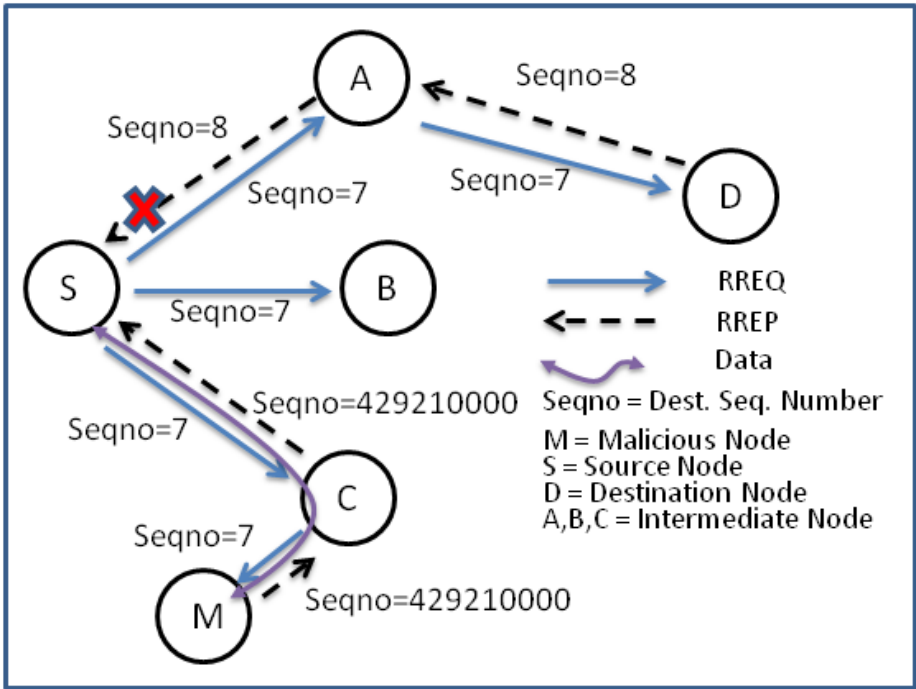


Fig. 3. Black hole attacks in AODV

3 Related Works in Detecting Black Hole Attack

There have been quite a number of works done in securing the routing protocol in MANET from the black hole attack. One example is by S. Yi [7] which looked at the Security-Aware Ad hoc Routing (SAR) using the security attributes such as trust values and relationships. Another example is the work done by Kimaya [8] which has proposed the used of Authenticated Routing for Ad hoc Networks (ARAN) i.e. a standalone protocol that uses cryptographic public-key certificates in order to achieve the security goals. Y.C Hu [9] on the other hand, has worked on the Secure Efficient Ad hoc Distance Vector Routing Protocol (SEAD) which employs the hash chains to authenticate hop counts and sequence numbers in the Distance Sequence Distance Vector (DSDV) protocol. Y.C Hu has also worked on Ariadne [10], which happen to be another secure routing protocol and it uses a shared secret key between two nodes based on the Dynamic Source Routing (DSR) Protocol.

The processing time involved in the above mentioned protocols has caused an extra overhead to the protocol. As a result, the protocols are less scalable and have delay. Although in most cryptographic method, the packets are normally authenticated, they are safe from external attacks, but in a black hole, the attacks are from inside where the packets are assumed authenticated, they can simply drop the packet passing through. Therefore, cryptographic methods cannot avoid such attacks.

Cryptographic method uses proactive approach but have some limitations. In order to address these limitations, a reactive approach is another way to trigger an action to protect the network from the malicious nodes. Zhang and Lee [11] present an intrusion detection technique for wireless ad hoc networks that uses cooperative statistical anomaly detection techniques. By using this technique, many numbers of false positives have been encountered. According to S. Lee in [12], the method requires the intermediate node to send Route Confirmation Request (CREQ) to the next hop towards the destination. This operation can increase the routing overhead which will then result in the performance degradation. In a related research, Stamouli [13] has proposed the architecture for Real-Time Intrusion Detection for Ad hoc Networks (RIDAN). The detection process relies on a state-based misuse detection system. As a result, each node would require extra processing power and sensing capabilities.

M.A. Shurman [14] in his work has proposed for the source node to verify the authenticity of the node that initiates the RREP messages by finding more than one route to the destination, so that it can recognize the safe route to the destination. This method can cause routing delay, since a node has to wait for a RREP packet to arrive from more than two nodes. Due to this, Dokurer [15] has proposed a solution based on ignoring the first established route to reduce the adverse effects of the black hole attack. His assumption is based on the fact that the first RREP message that arrived at a node normally would come from a malicious node. Unfortunately, this method has some limitations. For instance, the second RREP message received at a source node may also come from malicious node if the real destination node is nearer to the source node than the malicious node. This method also does not address on how to detect and isolate the malicious node from the network.

In a related work by N.R. Payal [16], the source node checks the RREP destination sequence number against a threshold value which is dynamically updated [17] at

every time interval. If the value is higher than the threshold, the RREP is suspected to be malicious. The ALARM packet will then be sent to the neighbours which contains the information of the black list (malicious) node as a parameter. An overhead of updating threshold value at every time interval along with the generation of ALARM packet will considerably increase the routing overhead. N.H. Mistry in [18] has proposed for the source node to verify the RREP destination sequence number by analysing the RREP messages which arrived within the predefined waiting period by using the heuristic method. If the sequence number is found to be exceptionally high, the sender of the respective RREP will be marked as malicious node. The major issue in this method is the latency time during the route discovery process since the source node has to wait until the waiting time period expired before the routing table can be updated. In the event where there is no attack in the network, the node still suffers with the latency time.

Generally, most methods discussed in [8] – [18], put some overhead on the intermediate and the source node. Therefore, the proposed algorithm for route discovery should consider the following objectives:

- Minimum routing overhead
- Low latency time
- Efficient processing.

4 ERDA: The Proposed Solution for Black Hole Attack

Based on the facts discussed in Section 2 and limitations highlighted in Section 3, a protocol which is the enhancement of the AODV is proposed. The ERDA (Enhanced Route Discovery AODV) is designed to improve the previous methods in terms of the overhead incurred during the route discovery. The proposed solution will employ minimum modification to the existing `recvReply()` function. There are three new elements introduced to improve the existing `recvReply()` function namely: the `rrp_table` to store incoming RREP packet, `mali_list` to keep the detected malicious nodes identity and the `rt_upd`, a new parameter to control the process of updating the routing table. Basically, the proposed method is divided into two parts, 1) Securing routing table update. 2) Detecting and isolating the malicious node. The Pseudo Code for the improved `recvReply()` function is shown in Figure 4.

4.1 Securing the Routing Table Update

In the conventional AODV as described in section 2, the node's routing table will be updated if it meets either one of these conditions a) the RREP's destination sequence number is higher than the one in the routing table or b) the destination sequence number in the RREP and the one in the routing table is equal but with less hop count. The proposed ERDA has imposed an additional condition by introducing a third parameter called `rt_upd`. This parameter can receive either true or false value. By default, the value is set to true which means the routing table is allowed to be updated although it does not comply with the existing conditions e.g. the destination sequence number in the RREP message is less than the one in the routing table.

In Figure 5, it explains how the ERDA works in the route discovery process and in the updating of the routing table. In the proposed ERDA, the route request (RREQ) message is sent out by the source node S to find a fresh route to the destination node D (as in the conventional AODV). All nodes that have “fresh enough route” including the destination node D will respond to that route request as shown in Figure 5(a). Every response received by node S will be captured into `rrep_tab`. Figure 5(b) shows the information contained in the `rrep_tab` Table for node S. The information contained in the table includes the `node_id` and also the destination sequence number. Since the malicious node M is the first node to respond with the RREP message to node S, the routing table of node S is updated with the information of node M as depicted in Figure 5(c). However, since the value of the `rt_upd` parameter in the ERDA is still ‘true’, it will allow the next RREP messages to update the routing table entry as well. Thus, when node S received the RREP message from node A, the message will be accepted although it arrived later and with a smaller destination sequence number than the one in the routing table. As a result, the former route entry will be overwritten by the later RREP coming from node A. The result is shown in Figure 5(d). The proposed ERDA offers a simple solution by eliminating the false route entry and replaced the entry path from the legitimate destination node. Since the later RREP comes from the actual destination node D, the `rt_upd` parameter value is then set to false. Any RREP message that comes after this point will be denied from accessing the routing table until the process of detecting malicious node is completed.

```

ERDA
1  RecvReply(Packet P) {
2    save P.srcIP and P.dst_seqno to rrep_tab
3    if (rt_upd = false) {
4      detect malicious node and save in mali_list
5      flush rrep_tab
6      set rt_upd to true
7    }
8    if (P.dst no entry in Routing Table RT) {
9      Add entry of P.dst to RT
10   }
11   select dst_seqno from RT
12   if (rt_upd = true) or
13   (P.dst_seqno > RT.dst_seqno or
14   P.dst_seqno = RT.dst_seqno and P.hops < RT.hops) {
15     if (P is from request destination node)
16       set rt_upd to false
17     update RT entry with P
18     send data packets to the route in RT
19   } else if (routing is UP for P) {
20     forward packet P
21     else discards P
17   }
18 }

```

Fig. 4. Pseudo code for the new `recvReply()` function

4.2 Detecting and Isolating the Malicious Node

The RREP's information like node id and destination sequence number is saved in the rrep_tab table as pointed out in section 4.1 above. The RREP message is allowed to flow as in the normal AODV. In the ERDA, the process of updating the route entry will continue until the value of rt_upd parameter is set to 'false' after receiving the RREP message from the destination node D. Later, the information in the rrep_tab table will be analysed using the heuristic method whereby the node id which has exceptionally high destination sequence number will be suspected as a malicious node [18] and the identity of those suspected nodes will be kept in the mali_list list. As a consequence, all nodes listed in the mali_list will be isolated from participating in future routing updates. Any control messages (e.g. RREP or RREQ) that come from those nodes also will be discarded in the network. In order to ensure that this process consumes less memory, the rrep_tab table will be flushed and the rt_upd parameter value will be set back to 'true' once the process of identifying malicious node is completed.

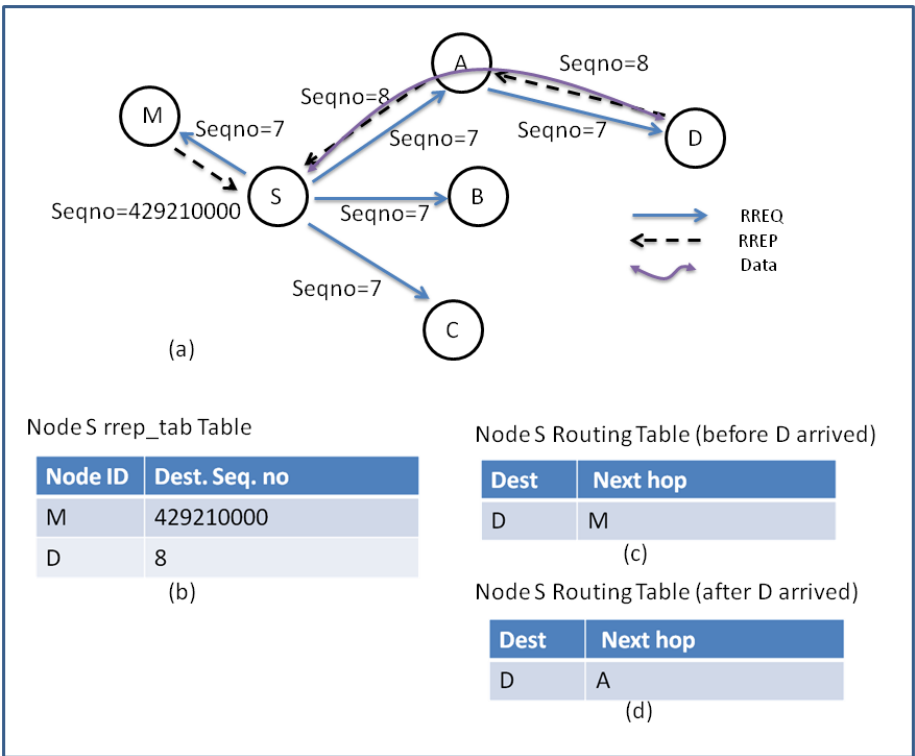


Fig. 5. Routing update in the ERDA.

5 Conclusions and Future Work

In this paper, the issue of black hole attack on the AODV-based routing protocol in MANET and also various methods to overcome this problem has been discussed. In our study, the route discovery process in the AODV is susceptible to black hole attack and therefore, it is vital to have an efficient security functions in the protocol in order to avoid such attacks. Based on the limitation of the previous methods, we presented our novel method, the ERDA, to prevent, detect and isolate the routing attacks in MANET.

ERDA is an enhancement of `recvReply()` function in the AODV protocol by improving the process of updating the routing entry and constructing a simple mechanism to detect and isolate malicious node without introducing significant delay during route discovery process. The enhancement only involves a minimum modification and does not change the existing protocol in the conventional AODV. Moreover, the ERDA does not incur high cost in terms of routing overhead and time overhead. The ERDA can resolve the black hole attack in the AODV-based MANET efficiently.

As future work, we intend to perform a simulation to test the ERDA method and to compare the result with the previous methods described in Section 3. However there are still research areas remaining which have to be examined in the future. 1) Exploring a new method for the ERDA to identified malicious node based on outlier detection algorithm [19][20]. 2) Performance and effectiveness of the ERDA to combat collaborative black hole and non-black hole attack in MANET. 3) Information sharing protection and privacy preservation in MANET using trusted ERDA.

References

1. Perkin, C.E., Royer, E.M.: Ad-hoc on demand distance vector routing. In: Proceedings of 2nd IEEE Workshop on Mobile Computer Systems and Applications, New Orleans (1999)
2. Abolhasan, M., Wysocki, T., Dutkiewicz, E.: A review of routing protocols for mobile ad hoc networks. Elsevier, Amsterdam (2004)
3. Mahmood, R.A., Khan, A.I.: A Survey on Detecting Black Hole Attack in AODV-based Mobile Ad Hoc Networks. In: International Symposium on High Capacity Optical Networks and Enabling Technologies (2007)
4. Perkin, C.E.: Ad hoc On Demand Distance Vector (AODV) Routing. Internet draft, draft-ietf-manetaodv-02.txt (November 1988)
5. Kumar, V.: Simulation and Comparison of AODV and DSR Routing Protocols in MANETs, Master Thesis (2009)
6. Xing, F., Wang, W.: Understanding Dynamic Denial of Service Attacks in Mobile Ad hoc Networks. In: IEEE Military Communication conference, MILCOM (2006)
7. Yi, S., Naldurg, P., Kravets, R.: Security-Aware Ad hoc Routing for Wireless Networks. In: Proc. 2nd ACM Symp. Mobile Ad hoc Networking and Computing (Mobihoc 2001), Long Beach, CA, pp. 299–302 (October 2001)
8. Sanzgiti, K., Dahill, B., Levine, B.N., Shields, C., Elizabeth, M., Belding-Royer: A secure Routing Protocol for Ad hoc networks. In: Proceedings of the 10th IEEE International Conference on Network Protocols, ICNP 2002 (2002)

9. Hu, Y.-C., Johnson, D.B., Perrig, A.: SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad hoc Networks. In: Proc. 4th IEEE Workshop on Mobile Computing Systems and Applications, Callicoon, NY, pp. 3–13 (June 2002)
10. Hu, Y.-C., Perrig, A., Johnson, D.B.: Ariadne: A Secure On-Demand Routing Protocol for Ad hoc Networks. In: Proc. 8th ACM Int'l. Conf. Mobile Computing and Networking (Mobicom 2002), Atlanta, Georgia, pp. 12–23 (September 2002)
11. Zhang, Y., Lee, W.: Intrusion detection in wireless ad – hoc networks. In: Proceedings of the 6th annual international Mobile computing and networking Conference (2000)
12. Lee, S., Han, B., Shin, M.: Robust routing in wireless ad hoc networks. In: ICPP Workshops, p. 73 (2002)
13. Stamouli, I.: Real-time Intrusion Detection for Ad hoc Networks. Master's thesis, University of Dublin (September 2003)
14. Shurman, M.A., Yoo, S.M., Park, S.: Black hole attack in wireless ad hoc networks. In: ACM 42nd Southeast Conference (ACMSE 2004), pp. 96–97 (April 2004)
15. Dokurer, S.: Simulation of Black hole attack in wireless Ad-hoc networks. Master's thesis, AtılımUniversity (September 2006)
16. Raj, P.N., Swadas, P.B.: DPRAODV: A Dynamic Learning System Against Blackhole Attack In Aodv Based Manet. International Journal of Computer Science Issues 2, 54–59 (2009)
17. Kurosawa, S., Nakayama, H., Kat, N., Jamalipour, A., Nemoto, Y.: Detecting Blackhole Attack on AODV-based Mobile Ad Hoc Networks by Dynamic Learning Method. International Journal of Network Security 5(3), 338–346 (2007)
18. Mistry, N.H., Jinwala, D.C., Zaveri, M.A.: MOSAODV: Solution to Secure AODV against Blackhole Attack. International Journal of Computer and Network Security (IJCNS) 1(3) (December 2009)
19. Subramaniam, S., Palpanas, T., Papadopoulos, D., Kalogerakiand, V., Gunopulos, D.: Online Outlier Detectionin Sensor Data using Non-parametric Models. J.Very Large DataBases, VLDB (2006)
20. Hawkins, D.M.: Identification of Outliers. Chapman and Hall, London (1980)

Fast Handover Technique for Efficient IPv6 Mobility Support in Heterogeneous Networks

Radhwan M. Abdullallah, Nor Asilah Wati Abdul Hamid,
Shamala K. Subramaniam, and Azizol Abdullah

Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400, Serdang, Selangor, Malaysia
radwanmas@yahoo.com, (asila, shammala, azizol)@fsktm.upm.edu.my

Abstract. The management techniques for Mobile IPv6 between different wireless technologies are very important to complete the handover process with the least possible delay. In the fast handover, when a mobile node moves to another network, it needs to do handover operations. These operations have a severe impact on the handover latency. This paper proposes an Enhanced Advanced Duplicate Address Detection (EA-DAD) method in a heterogeneous mobile environment with the support of the MIH services. The proposed method rapidly provides a unique Ipv6 address for MNs. At the same time, the binding updates to home agent and correspondent node are to be performed from old access router. We anticipate that the EA-DAD rapidly presents unique Ipv6 addresses for MNs with a minimum of handover latency as well as small buffering and packet loss even at high speed movements.

Keywords: Vertical Handover; FMIPv6; MIH; Duplicate Address Detection.

1 Introduction

In IP mobility, there are some important issues related to management techniques between different wireless technologies, these techniques depend on handover operation speed. Mobile IPv6 (MIPv6) [1] is one of the mobility management solutions that can support mobility between different access routers belonging to the same technology. This is realized primarily through using Care of Address (CoA) to indicate the location of the Mobile Node (MN). Although the MIPv6 protocol has several promising features and presents an elegant method to support mobility, it has an inherent weakness. That is, during the handover, the MN remains for a short time is unable to connection because of link switching delay and IP protocol operations. Handover is the operation by which the MN maintains an active connection as much as possible when it leaves from old to new networks. The handover operation occurs when the MN changes its network point of attachment from an old access router to a new access router, and it should achieve three operations: movement detection, new CoA configuration, and binding update. The MN is unable to connection as usual during the handover time. The duration of this time which is called handover latency is very critical for real-time services. Furthermore, there are many applications which

are sensitive to the amount of time in which the MN remains disconnected while performing a handover operation. This disconnection and delay affects the time of packet delivery within limited period (e.g. VoIP and audio/video streaming applications).

In the MIPv6 [1], a common handover method has been proposed to support mobility. But the amount of time it takes to complete the handover in MIPv6 is so large, that is sensitive for many applications. To reduce the handover latency, two categories of protocols have been proposed. One focuses on the network infrastructure design [2], [3], [4], [5], [6], such as HAWAII [5]. The other focuses on the method to reduce latency by MN and Access Router (AR) themselves [7], [8], [9], [10], such as Fast MIPv6 (FMIPv6) [8].

The FMIPv6 protocol has been proposed by the Internet Engineering Task Force (IETF) [8]. In this paper, we focus on the design of FMIPv6 protocol with the support of the MIH services to achieve vertical handover. The FMIPv6 protocol has been proposed by the Internet Engineering Task Force (IETF). It comes to address the following problem [7], [8], [9], [10]: how to allow an MN to send packets as soon as it detects a new subnet link, and how to deliver packets to an MN as soon as its attachment is detected by the New AR (NAR) [11]. It reduces the movement detection latency by anticipating handover and preparing the MN with the information about the NAR before disconnection from the current access router. This information is used to generating and configuration a new CoA by MN itself that is used to connect with a NAR. The typical address configuration requires Duplicate Address Detection (DAD) [12] to check validity of the configured address. To reduce the binding update latency, a bidirectional tunnel between Old Access Router (OAR) and the NAR is established.

However, it is noted that the DAD procedure easily takes up to 1 second, especially if the DAD begins after the link is created to the NAR. If duplicate CoA occurs, the handover latency will increase greatly. At the same time, we note that the binding updates to the Home Agent (HA) and Correspondent Node (CN) are performed after the time point when the MN is IP-capable on the new subnet link [8]. Because of this, the MN communicates with the CN directly via the NAR without using tunnel in a very late time. Thus the packet delay for some packets sent during the handover will be enlarged.

The IEEE 802.21, Media Independent Handover (MIH) [13] provides generic link-layer intelligence and other network related information to upper layers to optimize handover between different heterogeneous access networks like WiFi, WiMAX, and UMTS.

This paper proposed an Enhanced Advanced Duplicate Address Detection (EA-DAD) method with the support of the MIH services to reduce the latency vertical handover. To reduce the latency in the DAD procedure, we let the NAR generates and perform DAD for new CoAs and exchange groups of this address with the neighbor ARs and store this new CoA to the table before to any handover operation. Then when the MN requests the new CoA through the OAR, this new CoA will be distributed to the MN from the OAR. At the same time, to reduce the registration latency in the binding update, the binding update to the HA/CN will be performed by OAR. Only a small handover latency, buffering, and packet loss are expected when applying the EA-DAD, even with high speed movement.

The rest of this paper is organized as follows: section 2 discusses several related works. Section 3 describes the main idea of the proposal, EA-DAD with the support of MIH services. The conclusion is presented in section 4.

2 Related Work

2.1 FMIPv6

When the handover operation happens by changing between different networks, the MN should perform layer 2 handover and Layer 3 Handover. Layer 2 handover is the process with which the MN changes from one access point to another. Layer 3 handover is the process that the MN changes the attachment from one access router to another. The layer 3 handover includes three operations: movement detection, new CoA configuration, and binding update. Therefore the handover latency includes three parts:

- L2 handover latency, which is defined as the time interval from the moment that layer 2 link down trigger from the OAR happens to the moment that the layer 2 link up trigger to the NAR happens.
- Rendezvous time delay [14], which includes two kinds of latency. One is movement detection latency, which is the time interval taken by the MN to detect the presence of a NAR at the new access network, and the other is the new CoA configuration latency, which is the time interval taken to configure a new CoA for the MN.
- Registration delay, which is the time that it takes to send BU to the HA/CN and the subsequent resumption of communications indicated by a new data packet arriving at the MN from the NAR without passing through the tunnel between NAR and OAR.

To reduce the handover latency occurred in MIPv6, FMIPv6 are proposed in [8] that allows an AR to offer services to an MN in order to anticipate an IP layer handoff. The standard specifies two modes of operation namely Predictive and Reactive modes. The major difference between these two modes is on the time to establish the tunnel between the OAR and NAR. The Predictive mode of FMIPv6 is depicted in Figure 1. By the predictive mode, the tunnel is established before L2 handover, but by the reactive mode, the tunnel is established directly after L2 handover. We focus on the predictive fast handover method in this paper because it has shorter latency than the reactive one.

Although FMIPv6 was designed to overcome the major drawbacks of MIPv6 handoff, we noted that if the DAD is not performed at the beginning and is performed after the node has set up a link to the NAR, much time will be wasted if duplicate address exists and such latency cannot be tolerated for latency sensitive applications. On the other hand, we notice that the Binding Update (BU) cannot be started as soon as possible. By the predicative mode, the BUs to HA/CN are performed after the link is created to the NAR. After investigation, we find that when the handover starts and

the new CoA is known by OAR, the handover procedure will definitely be performed even if there are duplicated addressed or some other situations. Therefore, using this discovery, we propose performing the BU to HA/CN beforehand.

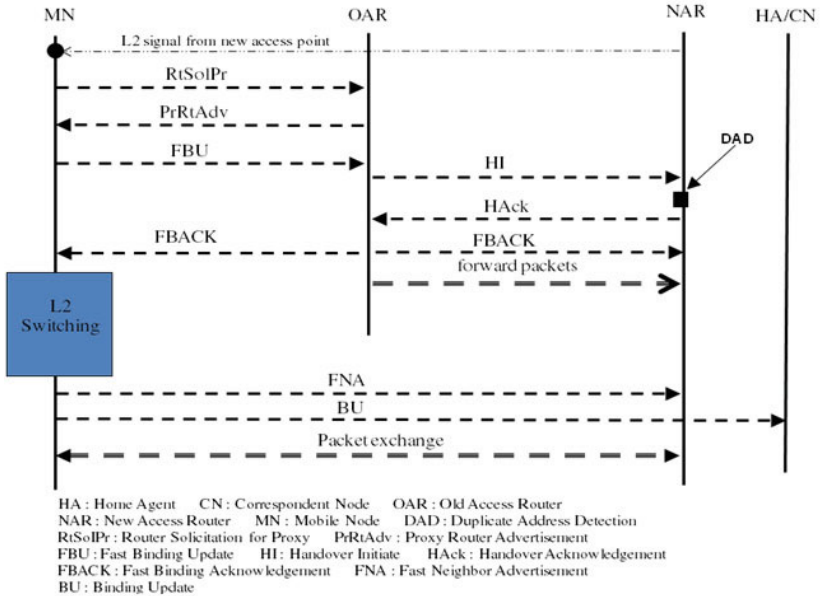


Fig. 1. Handover procedure in predictive FMIPv6

2.2 Media Independent Handover Services

In order to have a seamless handover, it's discovered that the timely information exchanges accurately for characterizing the network conditions is needed in order for appropriate actions to be taken [15]. Hence, IEEE recently published the IEEE 802.21 Media Independent Handover (MIH) services standard to enhance handovers across heterogeneous networks. The 802.21 does not implement network handover, relatively it provides information to allow handover to and from a range of networks including cellular, GSM, GPRS, WiFi, Bluetooth. The network handover enabling function within the protocol is implemented through the MIH function (MIHF). The MIHF consists of three elements, the Media Independent Event Services (MIES), the Media Independent Command Services (MICS), and the Media Independent Information Services (MIIS).

MIES provides event reporting corresponding to dynamic changes in link quality, link status, and link characteristics. The MICS enables users to manage and control link behaviors. The MIIS provides detail information about serving and neighboring networks. It will indicate which candidate network is suitable for the user to handover.

With IEEE 802.21 MIH services, the MN and the FMIPv6 network entities, in particular the AR, are informed about the values of the relevant parameters necessary

in handover decision making prior to the actual handover process. Furthermore, intelligent handover decisions to optimal subnets can be made with collaboration between the MN and the network entities. Thus, the MIH services enhance network discovery, preparation, and selection. The IEEE 802.21 assisted FMIPv6 scheme exploits the services of the MIHF, in particular MIIS to reduce handover delay, for example, the discovering and selecting the prospective network prior to actual operation of vertical handover.

2.3 Integration of FMIPv6 and IEEE 802.21

Even though FMIPv6 and IEEE 802.21 both aim to improve handover in different aspect, FMIPv6 barely deals with link layer mechanisms and focuses mostly on IP-layer messages. Hence, the media heterogeneity is not considered as an issue of link event detection. In contrast, IEEE802.21 gives a solution to vertical handoff while it deals mostly with media independency and improving layer-2 mechanisms.

Integrating these two protocols to optimize vertical handoff has been the interest of few proposals. Inadequacy of MIES primitives was the motivation of the work proposed in [16], [17] to create new primitives and use a handover mechanism similar to FMIPv6. Although these works originally address the issues of anticipation in FMIPv6, the method for AR discovery was not indicated and neither was information gathering from the neighborhood. In addition to the cost of improving L2 primitives, the proposed handover mechanism leaves more deployment complexities in AR.

Another solution was suggested in [18] through a protocol for discovering AR information called Access Router Information (ARIP); this approach is based on IETF SEAMOBY working group project [19] defined as Candidate Access Router Discovery (CARD). The method uses Radio Access Network Discovery (RAND) protocol to propose a complete layer-2 solution while still involving layer-3. The method is independent of mobility management and can be applied to both homogeneous and heterogeneous environments, as it does not rely on radio access technology. The information on neighboring ARs (ARIP) is provided at AR and sent to MN. However, the protocol suggests no method on how to collect ARIP info from AR and how the procedure should be initialized. In addition, maintaining such processes as described for AR requires more network resources together with more AR functionalities while some duties are still performed by MN. However, the protocol still requires AR deployment which is a technology obstacle.

Improving MIIS services was also the concern of few other proposals. Information in MIIS is specified in common formats and these information aid handoff decision-making process. Selecting a higher layer mechanism of mobility management to obtain information of neighboring networks from different access technologies is how MIIS information primitives are utilized in [20] and [21]. The higher layer mobility management is a SIP-based mechanism and the method was tested with an MN having two neighboring subnets to select. These approaches suggest the MIH information be obtained by MN through several query / response messages to estimate the network.

Vahid et al. [22] proposed solutions to improved Access Router Discovery (i-ARD) method to achieve a precise prediction of the prospective serving network. By this means, the message exchange between ARs is initiated using an established

tunnel, whereby neighboring ARs are queried for necessary information, and the mapping table at the serving AR is updated according to the MN needs. Although the handoff decision made in the network, MN cannot complete binding update with its HA and CN until its complete the connection with NAR. In addition, even if the procedure of the nCoA validation will start early this may be insufficient if the result of confirmation shows that the prospective nCoA is invalid (not unique).

2.4 Improved DAD

Many studies have been presented to eliminate or decrease the time required for DAD. The Advanced Duplicate Address Detection (A-DAD) [23] technique is one of these studies, and it improves on the DAD delay by storing a pool of unique IPv6 addresses at an AR. Each AR generates random addresses as a background process and performs a standard DAD on them. The addresses that are not duplicated are stored in the AR. The AR acts as a passive proxy for addresses. It listens to the neighbour discovery messages from other nodes in the network. If it hears another node performing DAD on the same address in its pool, the AR silently removes that address from its list and tests a new address to keep the list size constant. A-DAD is used with FMIPv6 as shown in Figure 2. It modifies FBU, HI, HAcK and FBacK messages with new options. The MN sends FBU with a new CoA-Request bit (R bit) to the OAR. When the NAR received HI messages containing the same R bit from the OAR, it immediately takes a unique IP address from its address pool and sends a HAcK message to the OAR. As soon as the MN receives such an FBacK from the OAR, it configures and confirms the new CoA. The time it takes for this process will be very short compared with the time for the standard DAD process.

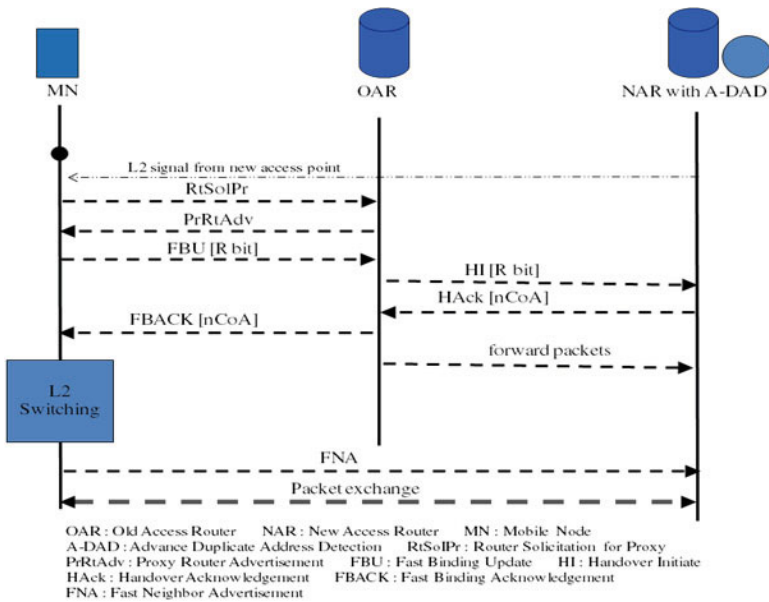


Fig. 2. A-DAD process with FMIPv6

Several other techniques are used to reduce the delay time to get a unique IP address. Optimistic DAD [24] works on the assumption that the probability of a duplicate address is very small. It allows the MN to use a Tentative Address in some communication while the uniqueness test is conducted by a normal DAD process. Proactive DAD [25] performs DAD on a new IP address before the MN moves to a new network. MLD-DAD [26] takes advantage of IPv6 multicast listener discovery. It assumes all nodes must join a solicited-node multicast address associated with each Tentative Address. So to check the uniqueness of an address, they simply verify whether a node is the first to enter the solicited-node multicast group when it is empty. This method could end up with false duplicates because multiple unicast addresses could join the same solicited node multicast group.

3 Proposed Method

Our Enhanced Advanced Duplicate Address Detection (EA-DAD) method will be based on the idea of stateful address configuration. We propose an improved A-DAD method to achieve a fast prediction of the prospective serving network. At the same time, we proposed enhanced FMIPv6 method by binding updates to HA/CN are brought beforehand.

3.1 Method Descriptions

EA-DAD similar to A-DAD but in a more efficient manner, EA-DAD separates the unique pool and redistributes it between neighbouring ARs. In more detail, the following steps should be performed beforehand.

- To reduce the DAD latency in FMIPv6 method, each AR randomly generates addresses as a background process.
- On generating a new address, each AR must perform its own DAD process to check the uniqueness of the address according to the standard [12].
- If the new addresses are unique, then the ARs that have overlaps in their coverage area will exchange groups of unique addresses and reserve them in the individual address table. These addresses are also considered as unique new CoAs.

When assigning an address to the MN, each AR that has this address must remove it from its table. The AR will generate a new address to keep the number of addresses in the table at a pre-defined value. Here, the new CoA will be generated and the DAD performed by AR beforehand to reduce the DAD latency. It is also important to note that except for the decrease on latency, bringing the DAD procedure ahead can create the issue of new unique CoA. Thus, the binding update to HA/CN can be brought forward because new CoA will no created.

Our method requires functionalities from the MIH services in form of discovering and selecting the prospective network prior to actual operation of vertical handover without requiring the whole 802.21 to be implemented. Also selecting the prospective network will help the OAR to exchange groups of the unique addresses with

prospective ARs. Through this advanced discovery method, a new entry for each discovered AR is created in the OAR and the table will be updated accordingly. Based on the selected information and the copy of the unique address of each network, the ARs are classified and sorted in the table.

3.2 Method Operation

The handover processes start as soon as the MN receives an L2 signal from one of the neighbouring networks. At that instant, the MN sends an nCoA-Req. message to the OAR including the L2 ID of the new network and a request option for a unique CoA. Upon receiving the nCoA-Req., by referring to the table, the OAR specifies the candidate network in which to handover the MN. Since the candidate AR is already specified, the OAR selects a unique CoA for the MN from the table depending on the new network prefix. It initiates a BU message for the unique address to the HA/CN and OAR also sends an nCoA-Adv. message to neighbouring ARs, which have the same table of address in order to remove the used addresses. The OAR replies to the MN with a nCoA-Rep. including the new CoA of the candidate AR which corresponds to the existing L2 ID.

As soon as the MN receives an nCoA-Rep. message, it configures the address specified by the new CoA into its second interface without any delay. At the time that the MN moves towards the new network, the LGD trigger will be issued from the MIH to inform the IP layer of the link as a result of signal degradation in the old interface. Simultaneously a LGD trigger is also issued from the MIH to the IP layer in

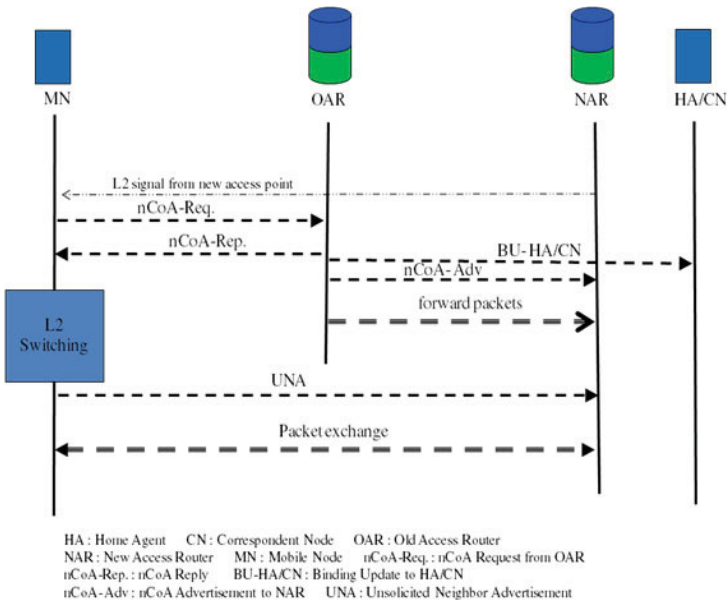


Fig. 3. Proposed handover method

the OAR. This event will start the tunnel establishment between the OAR and the NAR. The MN activates the second interface and establishes a low layer connection to the NAR while disconnecting the old connection. This L2 handover is performed through a Link switch (LS) command event which is received by the MIH layer at both the OAR and the MN. When the LU trigger is received right after the new connection is established, the MN is fully attached to the new network and sends an UNA message to the NAR to ensure the start of packet forwarding through the new connection as shown in Figure 3.

4 Conclusions

In this paper, we proposed a new fast handover method called EA-DAD in which the exchange messages have been reduced by providing unique IP address to the MN from the OAR. At the same time, we proposed that binding updates to HA/CN are performed by OAR. In our method we expect to reduce the handover delay as well as preventing disruption of service when the MN moves at a fast speed during the handover execution. We intend to simulate our approach to measure the handover latency, packet loss, and buffering size during the process of vertical handover.

References

1. Johnson, D., Perkins, C., Arkko, J.: Mobility Support in IPv6. IETF Network Working Group, <http://www.ietf.org/rfc/rfc3775.html>
2. Andrewt, T.C., Javieerg, G., Sanghyo, K., Andras, G.V., Chieh-Yan, W., Zoltan, R.: Design, Implementation, and Evaluation of Cellular IP. IEEE Personal Communication, 42–49 (2000)
3. Gustafsson, E., Jonsson, A., Perkins, C.E.: Mobile IPv4 Regional Registration. IETF Network Working Group, <http://tools.ietf.org/html/rfc4857>
4. Archan, M., Subir, D., Ashutosh, D., Anthony, M., Sajal, K.D.: IDMPbased Fast Handoffs and Paging in IP-based 4G Mobile Networks. IEEE Communication 40(3), 138–145 (2002)
5. Ramjee, R., Varadhan, K., Salgarelli, L., Thuel, S., Wang, S.Y., Porta, T.L.: HAWAII: a Domain-Based Approach for Supporting Mobility in Wide-Area Wireless Networks. IEEE ACM Trans. Networking 10(3), 396–410 (2002)
6. Soliman, H., Castelluccia, C., El-Malki, K., Bellier, L.: Hierarchical Mobile IPv6 Mobility Management (HMIPv6). IETF Mobile IP Working Group, <http://tools.ietf.org/html/draft-ietf-mobileip-hmipv6-08>
7. Blondia, C., Casals, O., Cerda, L., Wijngaert, N., Willems, G.: Performance Evaluation of Layer 3 Low Latency Handoff Mechanisms. Mobile Networks and Applications 9(6), 633–645 (2004)
8. Koodli, R.: Fast Handover for Mobile IPv6. IETF Network Working Group, <http://www.ietf.org/internet-drafts/draft-ietf-mobileip-fast-mipv6-08.txt>

9. Malki, K., Calhoun, P., Hiller, T., Kempf, J., McCann, P., Singh, A., Soliman, H., Thalanany, S.: Low Latency Handoffs in Mobile IPv4. IETF Network Working Group, <http://tools.ietf.org/html/rfc4881>
10. Velayos, H., Karlsson, G.: Techniques to Reduce the IEEE 802.11b Handoff Time. In: IEEE International Conference on Communications, vol. 7, pp. 3844–3848 (2004)
11. Dimopoulou, L., Leoleis, G., Venieris, I.: Fast Handover Support in a WLAN Environment: Challenges and Perspectives. *IEEE Network* 19(3), 14–20 (2005)
12. Thomson, S., Narten, T.: IPv6 Stateless Address Autoconfiguration, IETF Network Working Group, <http://tools.ietf.org/html/rfc4862>
13. Draft standard for Local and Metropolitan Area Networks: Media Independent Handover Services. IEEE P802.21/D10.0 (2008)
14. Vivaldi, I., Ali, B., Prakash, V., Sali, A.: Routing Scheme for Macro Mobility Handover in Hierarchical Mobile IPv6 Network. *IEEE Journal* 23(11), 2129–2137 (2003)
15. Yoo, S., Cypher, D., Golmie, N.: Timely Effective Handover Mechanism in Heterogeneous Wireless Networks. *Wireless Personal Communications Journal* 52(3), 449–475 (2008)
16. An, Y.Y., Yae, B.H., Lee, K.W., Cho, Y.Z., Jung, W.Y.: Reduction of Handover Latency Using MIH Services in MIPv6. In: 20th International Conference on Advanced Information Networking and Applications, pp. 229–234 (2006)
17. An, Y.-Y., Lee, K.W., Kum, D.W., Lee, S.H., Cho, Y.-Z., Yae, B., Jung, W.Y.: Enhanced Fast Handover Mechanism Using MIH Services in MIPv6. *Wired/Wireless Internet Communications*, 120–131 (2006)
18. Kassar, M., Kervella, B., Pujolle, G.: An Overview of Vertical Handover Decision Strategies in Heterogeneous Wireless Networks. *Computer Communications* 31, 2607–2620 (2008)
19. Liebsch, M., Singh, A., Chaskar, H., Funato, D., Shim, E.: Candidate Access Router Discovery (CARD)., <http://tools.ietf.org/html/rfc4066>
20. Mussabbir, Q.B., Wenbing, Y.: Optimized FMIPv6 Handover Using IEEE802.21 MIH Services. In: Proceedings of First ACM/IEEE International Workshop on Mobility in The Evolving Internet Architecture (2006)
21. Dutta, A., Das, S., Famorali, D., Ohba, Y., Taniuchi, K., Kodama, T., Schulzrinne, H.: Seamless Handover Across Heterogeneous Networks. An IEEE802.21 Centric Approach (2006)
22. Solouk, V., Ali, B.M., Khatun, S., Daniel, K., Adzir Mahdi, M.: Layer-2 Protocol Adaptation Method to Improve Fast Handoff for Mobile IPv6 Vertical Handoffs. *JCM* 4(6), 396–403 (2009)
23. Han, H., Hwang, H.: Care-of Address Provisioning for Efficient IPv6 Mobility Support. *Computer Communications Journal* 29(9), 1422–1432 (2006)
24. Moore, N.: Optimistic Duplicate Address Detection for IPv6. IETF Network Working Group, <http://tools.ietf.org/html/rfc4429>
25. Tseng, C., Wong, Y., Yen, L., Kai, H.: Proactive DAD: A Fast Address-Acquisition Strategy for Mobile IPv6 Networks, *IEEE* 10(9), 50–55 (2006)
26. Daley, G., Nelson, R.: Duplicate Address Detection Optimization using IPv6 Multicast Listener Discovery. IETF IPv6 Working Group, <http://tools.ietf.org/html/draft-daley-ipv6-mcast-dad-02>

Using Dendritic Cell Algorithm to Detect the Resource Consumption Attack over MANET

Maha Abdelhaq, Rosilah Hassan, and Raed Alsaqour

School of Computer Science, Faculty of Information Science and Technology,
University Kebangsaan Malaysia, 43600, UKM, Bangi, Selangor Darul Ehsan, Malaysia
maha@ftsm.ukm.my, rosilah@ftsm.ukm.my, raed.saqour@ftsm.ukm.my

Abstract. Artificial Immune Systems (AISs) and Mobile Ad Hoc Networks (MANETs) are two up to date attractive technologies. AIS is utilized to introduce efficient intrusion detection algorithms to secure both host based and network based systems, whilst MANET is defined as a collection of mobile, decentralized, and self organized nodes. Securing MANET is a problem which adds more challenges on the research. This is because MANET properties make it harder to be secured than the other types of static networks. We claim that AIS properties as robust, self-healing, and self-organizing system can meet the challenges of securing MANET environment. This paper objective is to utilize the benefits of one of the Danger Theory based AIS intrusion detection algorithms, namely the Dendritic Cell Algorithm (DCA) to detect a type of Denial of Service (DoS) attack called Resource Consumption Attack (RCA) and also called sleep deprivation attack over MANET. The paper introduces a Mobile Dendritic Cell Algorithm (MDCA) architecture in which DCA plugged to be applied by each MANET node.

Keywords: Mobile Ad Hoc Networks, Denial of Service attack, Resource Consumption Attack, Danger Theory, Artificial Immune System, Dendritic Cell Algorithm.

1 Introduction

Danger theory [1] is considered as one of the essential cores on which a number of Artificial Immune System (AIS) algorithms based to detect threats and intrusions. Danger theory implies that the concentration of the danger or safe signals which come from the body tissues and caused by specific antigens control the response of the Human Immune System (HIS) to tolerate or fight those antigens.

As - in biology- the danger comes from some kinds of viruses, fungus, parasites, and bacteria to affect harmfully on the human body. In computer systems, the danger comes also from different types of attacks to disrupt and destroy either host based or network based systems.

In 2003, Aickelin et al [2] came up with a project called “danger project” in order to support utilizing the danger theory in developing AIS intrusion detection algorithms. Dendritic cell algorithm (DCA) [3] is one of the most well-known danger project contributions. It utilizes the role of the dendritic cells (DCs) in HIS as forensic

navigators and important anomaly detectors. DCs are defined as antigen presenting lymphocytes in the innate immunity. These lymphocytes play the main role in either stimulating or suppressing the adaptive immunity T- cells. Hence, DCs control the immune system type of response either to tolerate or fight the antigens respectively.

DCA [4] proved the capability of detecting port scanning attack which certifies its qualification as an anomaly detector algorithm. This opens the way of using it to detect other types of attacks over securing challenging environments such as Mobile Ad hoc Network (MANET). As stated by Kim et al [5], the properties of danger theory based AIS intrusion detection algorithms, especially DCA can meet the security requirements of sensor networks. However, sensor networks are defined as one of MANET's types with more conditions and restrictions [6]. In addition, many of MANET's special characteristics and properties are similar to the innate immunity abstract features declared by Twycross and Aickelin [7]; such as the openness and susceptibility of each to different types of danger attacks. Therefore, we argue that DCA features can also meet the security requirements of MANET environment as a mobile, decentralized, limited power, and limited capacity wireless network. The objective of this paper is to explain the capability of the danger theory AIS intrusion detection algorithms, in particular DCA to detect a type of denial of service (DoS) attack called resource consumption attack (RCA) over MANET. Also, the paper introduces a mobile dendritic cell algorithm (MDCA) architecture in which DCA plugged to be applied by each MANET node.

This paper is structured as follows: Section 2 introduces a literature review of the danger theory based AISs. Next, section 3 shows the mapping between MANET and the artificial tissues proposed in the "danger project". After that, section 4 explains Ad hoc on Demand Distance Vector (AODV) routing protocol and its vulnerability to RCA. Next, section 5 explains how DCA is capable to detect RCA. Finally, section 6 represents a conclusion for this research with future work.

2 AIS Intrusion Detection Algorithms and Frameworks Overview

This section introduced a literature review of danger based AISs. Some of these AISs applied over MANET and some introduced to be applied over wired network. DCA has been extensively explained with its equations and the required weights.

2.1 Related Works

Sarafijanovic and Boudec [8, 9] introduced the first researches that utilized AIS to be applied on MANET. Their introduced intrusion detection architecture is applied over the network layer protocol of the Open System Interconnection model (OSI). They depend on a co-stimulation concept represented by a danger signal to inform about the packet loss on the connection path. The proposed AIS architecture consists of four main modules; the thymus module, the danger module, the clustering module, and the clonal selection module. The Thymus module performs negative selection operation; in negative selection lymphocytes that show strong bind with the self antigens are killed. The danger module produces the danger signal if no acknowledgment received for the sent packet. The clustering module is used to verify the detection. And the

clonal selection module used to enhance the detectors quality. The proposed AIS registered a detection rate of about 55% but the whole system could only detect a simple dropping packet attack.

Greensmith et al [3, 4] proposed a new DC-based Algorithm called DCA. The algorithm is considered as a main contribution in the danger project established by Aickelin et al [2]. Also, it is built over the libtissue architecture proposed by Twycross and Aickelin [10]. DCA is inspired from immunological researches on DCs because of their desired characteristics as mobile anomaly detectors. The algorithm was verified by applying it to detect port scanning attack over wired network [4].

Kim et al [5] used a theoretical integration between the DCA and directed diffusion routing protocol to protect the sensor network from interest cache poisoning attack.

Fanelli [11] proposed The Network Threat Recognition with Immune Inspired Anomaly Detection (NetTRIIAD) model. NetTRIIAD model utilized the danger theory and implements the negative selection in a different way than the previous self non-self based researches. The model consists of two main layers; the innate layer which emulates the innate immunity in HIS, and the adaptive layer which emulates the general abstract adaptive immunity in HIS. The innate layer collects required data to detect the intrusions, whilst the adaptive layer performs the negative selection algorithm. NetTRIIAD is used to detect DoS, dropping packets and delaying packets over wired networks. NetTRIIAD uses a correlation method between signals and its related antigens which reduces the false positive rates in the detection.

Drozda et al [12] use the concept of co-stimulation and communication between the innate immune system and the adaptive immune system to introduce an AIS intrusion detection algorithm. The algorithm detects three types of attacks over MANET: wormhole attack, dropping packets attack, and packet delay. If the first step of the introduced algorithm detects the existence of attacker, it stimulates the second step of the detection which is energy inefficient and used it for confirmation. Else, no need for that energy inefficient stage because it depends on overhearing the packets sent by the neighbour to the 2-hops neighbour (watchdog) [13]. A neural network mechanism is used to improve the algorithm's optimization. However, the algorithm has many drawbacks. First, the cooperation between each 2-hops neighbour in the detection causes traffic overhead. Second, the 2-hops neighbour may not be trusted. Finally, the algorithm depends mainly in its confirmation stage on watchdog which fails when a collision occurs, or the malicious node changes its power to make it includes the previous node but not the next one [13].

2.2 Dendritic Cell Algorithm

This paper is concerned in the DCA proposed by Greensmith et al [3, 4]. DCA consists of three main stages: initialization, update, and aggregation. In the initialization stage, the algorithm sets the important parameters and the update stage includes tissue update with DCs, antigens, and signals asynchronously. The DCs and signals are updated each one second not simultaneously. The antigens updated when it is available. After that, the antigens are transferred from the client tissue and stored in the server tissue to be processed in the DC cycle. The same done with updated signals as in the libtissue architecture [10]. One DC plays in each iteration of the algorithm

cycle. The DC exposes to a collection of input antigens and the available input signals which are in four main categories: (i) Pathogen Associated Molecular Patterns (PAMPs), (ii) Danger signals, (iii) Safe signals, and (iv) Inflammation.

The DC applies the concentration equation 1 using the input signals and empirically calculated weights in table 1 to calculate the concentration of three main DC outputs which are: Costimulatory molecules (*esm*), smDC cytokines (*semi*), and mDC cytokines (*mat*). The main indices and data structures of equation 1 are as follows:

- Indices:
 - $i = 0 \dots i$; input signal index;
 - $j = 0 \dots j$; input signal category index;
 - $m = 0 \dots m$; DC index;
 - $p = 0 \dots p$; DC output signal index;

● Data Structures

- S = tissue signal matrix; (the concept of “tissue signal” is explained in [4]).
- S_{ij} = a signal type i , category j in the signal matrix S ; (matrix S explanation Can be found in [4]).
- $S(m)$ = signal matrix of DC(m);
- $O_p(m)$ = output signal p of DC(m);
- W_{ijp} = transforming weight from $S_{ij}O_p$.

Table 1. Weights used to process the input signals in equation 1

W_{ijp}	PAMP	Danger Signal	Safe Signal
	J = 0	j = 1	j = 2
<i>esm</i> p = 1	2	1	2
<i>semi</i> p = 2	0	0	3
<i>mat</i> : p = 3	2	1	-3

$$O_p(m) = \frac{\sum_i \sum_{j \neq 3} W_{ijp} S_{ij}(m)}{\sum_i \sum_{j \neq 3} |W_{ijp}|} \forall P \tag{1}$$

When *esm* exceeds certain fuzzy threshold, the DC must migrate and the calculation is stopped. That threshold is user defined determined by equation 1. When that threshold is small, the antigens are not exposed to enough amounts of signals. And the advantage of the algorithm is done when it is large enough to expose the antigens to signals for a longer time which leads to more accurate results. When the DC migrates, the calculated concentration of semi mature and mature output signals are compared and the DC given the context of the larger value. Hence, Each DC context is given to the whole collection of antigens exposed to certain signals into that DC in a certain fuzzy threshold time. After migration, the update is done again and a new DC begins its work with new population of antigens and signals until the whole number of input items (antigens) are processed. At the end of the algorithm, the aggregation stage registers each antigen context value in a log file. And hence, the degree of maturation for each antigen is calculated as in equation 2 as follows:

$$\text{Mature Context Antigen Value (MCAV)} = \frac{\text{the number of times the antigen appeared as mature}}{\text{the number of times it appeared in the log file}} \quad (2)$$

3 Artificial Immune Systems and MANET

AIS intrusion detection algorithms aim at getting benefits from the HIS subsystems by mapping their functions and concepts in biology into abstract artificial frameworks. Mapping between HISs and AISs is an enjoyable challenge which needs testing and verification. This section sheds light on introducing MANET as a technology capable to be an application for danger theory based AIS algorithms.

3.1 Mobile Ad Hoc Network Overview

MANET is a rapidly deployable, self-organized, multi-hop wireless network, and typically set up for a limited period of time and for particular applications such as military, disaster areas, and medical applications. Nodes in MANET may move arbitrarily while communicating over wireless links. This network is typically used in situations where there is no centralized administration or support from networking infrastructure such as routers or base stations. Thus, nodes must act as both routers, end-systems and organize themselves into a wireless network. Many up to date researches pay attention to work on MANET as a new technology with specific characteristics which distinguish it from other types of networks, these characteristics are as follows [6, 14]:

C1: Openness – MANET nodes communicate with each other through an open wireless medium. Hence the outer attackers can easily join themselves with in the nodes environment.

C2: Limited resources – MANET has limited power and bandwidth capacity.

C3: Mobility and Dynamicity - MANET consists of highly frequently mobile nodes which causes high dynamicity in its topology changes and reconfiguration.

C4: Wireless medium signaling - the nodes in MANET interacts with each other through wireless signaling.

C5: Flexibility – MANET could be deployed in any types of areas even if they are unstable such as military purposes areas, or the areas of frequent nature disasters.

C6: Decentralization and self-organizing – MANET is an infrastructure less wireless network with no centralized management points, so every node manages itself by itself and can help managing the other nodes but with no centralization such as sending alarm messages when an attacker detected.

C7: Distributed Computation – each node performs a routing processing, a security processing and inform the other nodes to help survive the network.

3.2 The Analogy between MANET and the Innate Immunity Framework

The innate immunity in biology has an important role in detecting danger coming from outside to invade the human body. It consists of forensic navigator cells which navigate throughout the body tissues to protect them from dangerous pathogens. The innate immunity cells as mobile, self-organizing, and flexible cells inspire showing the analogy between MANET environment special characteristics and the abstract properties of the innate immunity environment based on the work of Twycross and aickelin [7]. The following displays the general innate immunity properties and the corresponding MANET characteristics mentioned in the previous subsection:

- Innate immunity cells navigate through tissues susceptible to an open environment full of different types of invaders coming from outside the human body, the same as MANET environment characteristic C1.
- Each innate immunity cell has a limited capacity so each cell process a limited amount of proteins also it communicates with a limited number of neighbor cells. The same as MANET characteristic C2.
- Innate immunity cells move and organize themselves and can navigate throughout different types of body tissues in a flexible decentralized manner which reflects MANET characteristics C3, C5, and C6.
- Innate immunity cells interact with the adaptive immunity cells through chemical cytokines signals, MANET nodes communicate with each other through wireless signaling C4.
- Innate immunity cells perform computational processing for the incoming proteins in parallel, in order to help survive the human body the same as what the MANET nodes do in characteristic C7.

4 AODV and Its Vulnerability to Resource Consumption Attack

AODV routing protocol [15] is the underlying routing protocol used in this research. In abstract, AODV is a reactive self-starting, and large scale routing protocol. AODV routing protocol has been extensively studied and developed over many years which proves its robustness and benefits. The main advantages of this protocol are firstly, the connection setup delay with the destination is lower comparing with other MANET routing protocols. Secondly, AODV avoids the congested paths in comparison with the other ad hoc routing protocols. Thirdly, it supports both unicast and multicast communications. Fourthly, it can cope with the rapid ad hoc topological reconfigurations that may affect the other routing protocols [16]. However, AODV is vulnerable to different types of attacks. The following subsections explain AODV processes and how it is vulnerable to RCA over MANET.

4.1 AODV Routing Protocol

In the route discovery process of AODV routing protocol over MANET [15], the source node broadcasts route request (RREQ) packet throughout MANET nodes -as shown in Fig. 1- and set a timer waiting for reply. The RREQ packet contains routing information such as: the originator IP address, the broadcast ID, and the destination sequence number.

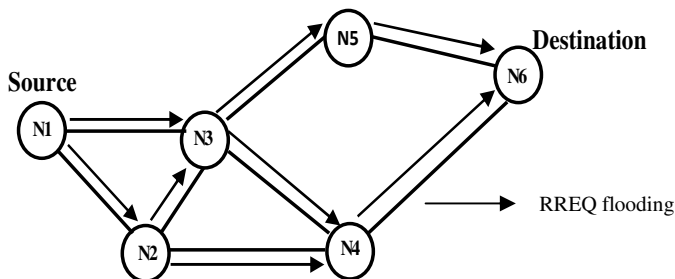


Fig. 1. Propagation of RREQ packet

Each intermediate node receives the RREQ packet and keeps the reverse path to the source node besides performing two processes: firstly, it verifies if it receives the RREQ packet before with the same originator IP address and broadcast ID, then decided either to discard the RREQ packet or accept it. This verification process helps preventing flooding attack such as RCA. Secondly, if the RREQ packet is accepted, the intermediate node checks the destination sequence number stored in its routing table, if it is greater than or equal to the one stored in the RREQ packet it unicasts the route reply (RREP) packet to source node. If no intermediate node has fresh enough (fresh destination sequence number) route to the destination node, the RREQ packet keeps its navigation until it reaches the destination node itself which in turn unicasts the RREP packet towards the source node as shown in Fig. 2.

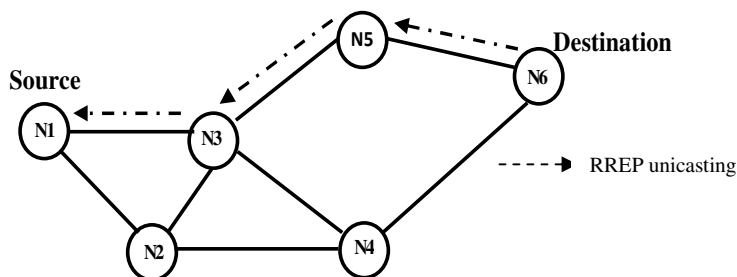


Fig. 2. The path of RREP packet

4.2 Resource Consumption Attack over AODV

RCA [17] is one of the DoS attacks in which the attacker exploits the route discovery process in AODV routing protocol. The attacker as -shown in Fig. 3- keeps broadcasting RREQ packet with different broadcast ID in order to notify each node continuously processing the RREQ packet and consume its limited resource of energy, bandwidth, and memory.

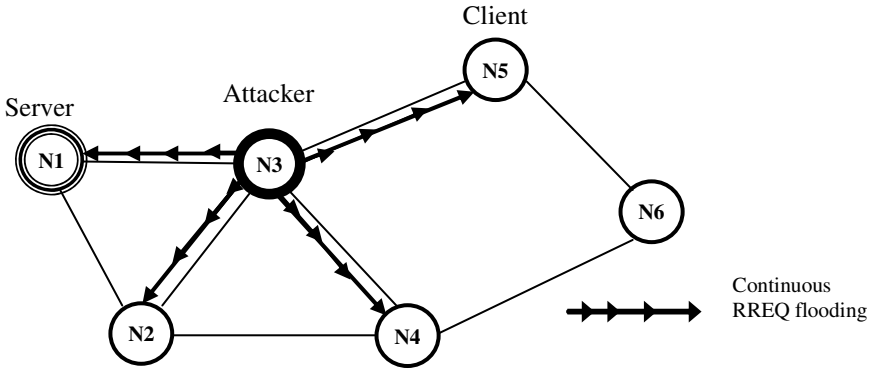


Fig. 3. RREQ broadcasted by the Resource Consumption Attacker

As noticed, the attacker does not follow AODV rules therefore, to achieve it’s attacking successfully, it does not set a timer waiting for reply but keeps overflowing the network with RREQ packets as shown in Figure 4. MANET is very vulnerable to this type of attack since its limited bandwidth capacity simplifies overflowing the link very easily and quickly. When MANET links have over flown with malicious packets, the links will be jammed and congested which leads to interrupt accessing services of the available servers in the network. In Figure 4 if node N1 represents a server, then its service could be isolated by the attacker N3.

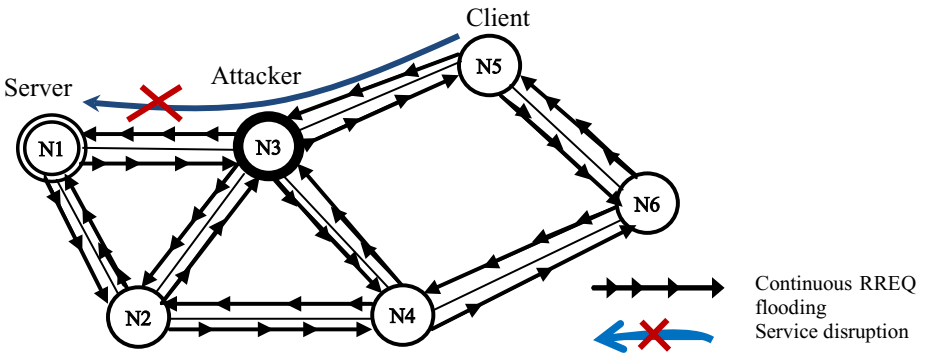


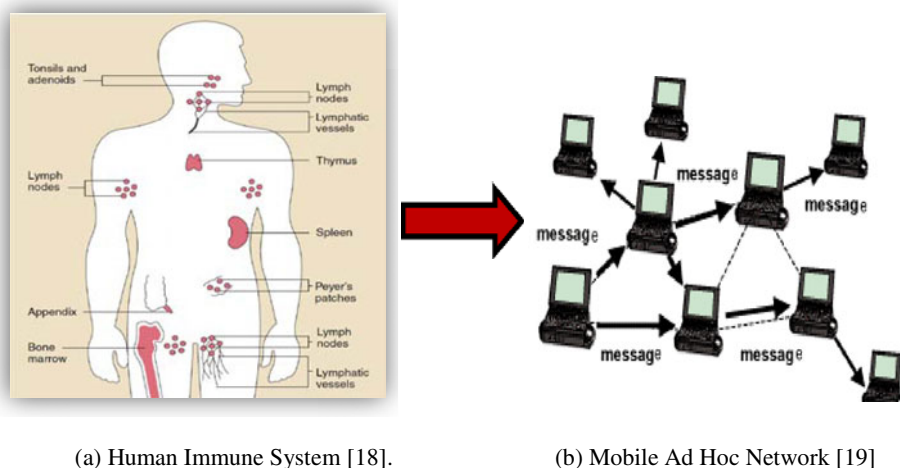
Fig. 4. MANET flooded with RREQ packets by RCA attacker

5 Using DCA to Detect the Resource Consumption Attack

As mentioned in section 3, many properties are shared between MANET and the innate immune system, one important property is that the two environments are open and vulnerable to danger either from outside or inside. All of the sharing features and the environment nature encourage utilizing the danger based AISs which are abstracted their functionality from the innate immunity and its cells. Dendritic cells are one of the innate immunity cells which inspired developing a danger based AIS intrusion detection algorithm called DCA. The following subsections show how DCA could be effective in detecting RCA over MANET.

5.1 The Proposed System Architecture

As shown in Fig. 5, a mapping between HIS and MANET is performed in general. For example, each message in MANET represents the entered pathogen to the human body. Each node represents the human body or part of the human body. Therefore, each node must apply MDCA to protect itself from intrusions same as each human body contains the immune system to protect itself from dangerous pathogens. As shown in Fig. 6, DCA has been plugged in a new architecture called Mobile Dendritic Cell Algorithm (MDCA) architecture. The name shows that the proposed architecture will be performed by each MANET node since each node should protect itself from danger locally without using mobile agents.



(a) Human Immune System [18].

(b) Mobile Ad Hoc Network [19]

Fig. 5. Mapping HIS model into AIS algorithm over MANET

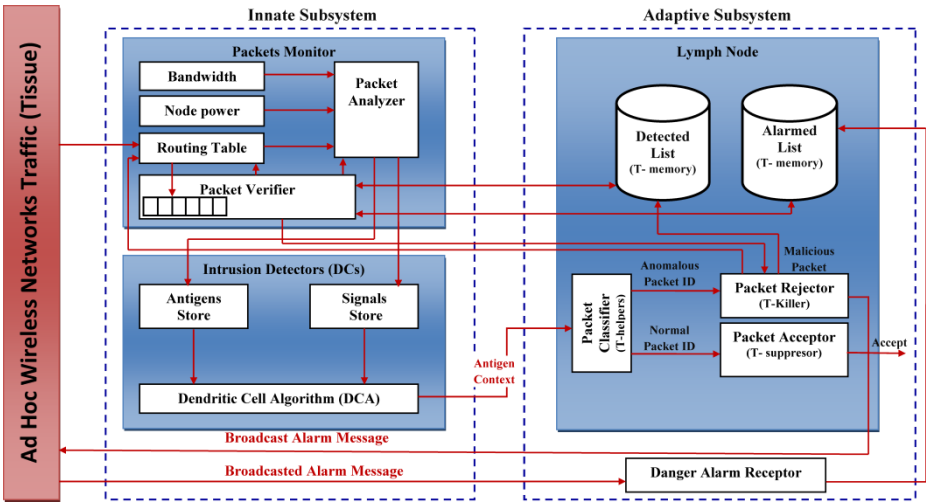


Fig. 6. The Proposed MDCA architecture

MDCA architecture consists of two main subsystems; innate subsystem and adaptive subsystem. The architecture represents a conceptual mapping of MDCA pseudo code algorithm shown in Fig. 7. At the beginning, the algorithm verifies each entered packet’s ID in the memory. If that packet ID found in the detected list, this means it comes from an attacker detected before, the algorithm rejects the packet directly, delete its information from the routing table and sends an alarm message for the second time for that packet ID. Else if the packet ID is found in the alarmed list, this means the packet comes from an attacker detected by another node so it is rejected directly, delete its information from the routing table but without sending alarm again. Else, the packet must be analyzed by the packet analyzer.

The packet analyzer extracts the required antigens from the routing table and generates the signals from the routing table, availability of the bandwidth, and the power consumption rate. After that, the packet analyzer stores the antigens and signals in the antigens and signals stores respectively. The available antigens and signals formulate the main input for DCA to detect the RCA attacker over MANET. DCA, which utilizes the abstract functionality of DCs in biology, performs the intrusion detection exactly as mentioned in subsection 2.2. DCA outputs of antigens and their corresponding contexts as *benign* or *malignant* transfer to the packet classifier in the adaptive subsystem. The packet classifier plays the abstract role of T-helpers cells in the HIS therefore; if the output antigen context is *benign* it will suppress any fighting reaction against the antigen by transferring it to the packet acceptor same as the abstract role of T-suppressor cells. Else, if the output antigen context is *malignant*, the packet classifier will stimulate fighting that antigen by transferring it to the packet rejector unit which simulates the abstract role of T-killer cells in HIS.

```

Input: traffic packets
Output: classified packets as normal or malicious
Store input packet ID in Queue
While packet queue!= null
|   get packet ID
|   verify packet ID in memory
|   if packet ID exist in detected list
|   |   reject packet
|   |   delete the packet info from the routing table
|   |   broadcast alarm message
|   else if packet ID exist in alarmed list
|   |   reject packet
|   |   delete the packet info from the routing table
|   else
|   |   extract packet antigens
|   |   transfer packet antigen to antigen store
|   |   extract packet signals
|   |   transfer packet signals to signal store
|   |   call DCA
|   |   if antigen is benign
|   |   |   accept the packet
|   |   |   start the routing algorithm
|   |   else reject the packet
|   |   |   delete the packet info from the routing table
|   |   |   broadcast alarm message
|   |   |   store packet ID in detected list
|   |   end if
|   end if
|   end if
end while loop

```

Fig. 7. The proposed MDCA pseudo code

Benign antigens represent the packet ID of normal nodes and vice versa, *malignant antigens* represent the packet ID of the malicious nodes. The normal packets are accepted. But the anomalous packets treated differently, the packet rejecter rejects these packets, deletes them from the routing table, registers them in the detected list, and sends an alarm message for the whole neighbor nodes to inform them about the attacker. Finally, MDCA architecture as shown in Fig. 6 contains a danger alarm receptor that receives the alarms come from the neighbor nodes and registers them in the alarmed list to isolate the attacker and prevent it after detecting its anomalous behaviors.

5.2 Antigens

In HIS, antigens and tissue signals are two important inputs for DCs to control T-helpers response; either to fight the *malignant antigens* or suppress fighting the

benign ones. Signals represent the symptoms of danger or safe state existence. However, antigens represent the resource of danger or safe state.

According to MDCA over MANET, the resource of normal or anomalous behaviors is the mobile nodes themselves. Identifying the resource node of danger helps preventing it forever by isolating it from the network. Therefore; MDCA considers the antigen to be the IP address of the RREQ packet originator. By this way, MDCA could perform two types of responses: firstly, it detects the danger very early especially when the same attacker comes again. Secondly, it prevents the attack in the whole network by broadcasting the IP address of the malicious node in alarm messages throughout the network.

5.3 Signals

DCA as mentioned in subsection 2.2 includes four main input signals that specify the behavior of the input antigens. This paper utilizes three input signals only: (i) PAMP signal, (ii) Danger signal, (iii) Safe signal, and the Inflammation input signal has not yet utilized in this paper. The details of MDCA signals are as follows:

- *High rate of the received RREQ control packets by the routing table (PAMP)*: the abnormal increase in the received rate of RREQ control packets by the routing table indicates strongly the existence of resource consumption attack. The routing table supports the packet verifier in MDCA architecture with this information as a PAMP signal to the available antigens.
- *Abnormal rate of the battery power consumption (danger signal)*: if the node's battery losses its power in abnormal rate, this indicates the success of RCA. However, this signal absence does not mean necessarily the absence of the attacker; since at the beginning of the attack the high rate of the received RREQ packets is only noticed.
- *Failure in routing discovery and data packets delivery (danger signal)*: when the attacker overflows the wireless links with bogus RREQ packets, it becomes congested and flooded easily and quickly because of its limited bandwidth. This problem causes failure in both routing discovery and data packets delivery.
- *Success in routing discovery and data packets delivery (safe signal)*: taking into consideration to process the safe signals in parallel with the other signals decreases the false positive rate in the intrusion detection algorithm. However; the existence of these signals does not improve mainly the absence of attack. If the node succeeded in initiating its routes and communicates with the other nodes freely, this means -in somehow- the failure of RCA attacker(s).

6 Conclusion and Future Work

This paper has utilized the benefits of one of the Danger Theory based AIS intrusion detection algorithms called DCA to detect the resource consumption attack over MANET. DCA has been plugged into a new mobile intrusion detection and prevention architecture called MDCA architecture. MDCA has to be performed by each node in MANET to detect the attack locally without any need for mobile agents.

MDCA will be verified and tested by performing simulation experiments in the future work. However, in the future experiments *csm* fuzzy threshold should be determined in a good way which avoids the research to fall into high false positive rates. Also, MCAV thresholds should be determined in order to determine the context of the tested antigen if it is *benign* or *malignant*. Finally, more signals and antigens will be added to enhance the intrusion detection precision and decrease the possible false positive rates.

Acknowledgment

The three authors are members in the Network and Communication Technology (NCT) group, University Kebangsaan Malaysia. [http:// www.ftsm.ukm.my/network](http://www.ftsm.ukm.my/network).

References

1. Matzinger, P.: Tolerance, Danger, and the Extended Family. *Annual Review of Immunology* 12, 991–1045 (1994)
2. Aickelin, U., Bentley, P., Cayzer, S., Kim, J., McLeod, J.: Danger Theory: The Link between AIS and IDS? In: Timmis, J., Bentley, P.J., Hart, E. (eds.) ICARIS 2003. LNCS, vol. 2787, pp. 147–155. Springer, Heidelberg (2003)
3. Greensmith, J., Aickelin, U., Cayzer, S.: Introducing Dendritic Cells as a Novel Immune-Inspired Algorithm for Anomaly Detection. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, pp. 153–167. Springer, Heidelberg (2005)
4. Greensmith, J., Aickelin, U., Tedesco, G.: Information Fusion for Anomaly Detection with the Dendritic Cell Algorithm. *Information Fusion* 11, 21–34 (2010)
5. Kim, J., Bentley, P.J., Wallenta, C., Ahmed, M., Hailes, S.: Danger Is Ubiquitous: Detecting Malicious Activities in Sensor Networks Using the Dendritic Cell Algorithm. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 390–403. Springer, Heidelberg (2006)
6. Cayirci, E., Rong, C.: *Security in Wireless Ad Hoc and Sensor Networks*. Wiley, United Kingdom (2009)
7. Twycross, J., Aickelin, U.: Towards a Conceptual Framework for Innate Immunity. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, pp. 112–125. Springer, Heidelberg (2005)
8. Sarafijanovic, S., Le Boudec, J.Y.: An artificial immune system for misbehavior detection in mobile ad-hoc networks with virtual thymus, clustering, danger signal and memory detectors. In: Nicosia, G., Cutello, V., Bentley, P.J., Timmis, J. (eds.) ICARIS 2004. LNCS, vol. 3239, pp. 342–356. Springer, Heidelberg (2004)
9. Sarafijanovic, S., Le Boudec, J.Y.: An artificial immune system approach with secondary response for misbehavior detection in mobile ad hoc networks. *IEEE Transactions on Neural Networks* 16(5), 1076–1087 (2005)
10. Twycross, J., Aickelin, U.: Libtissue - Implementing Innate Immunity. In: *IEEE Congress on Evolutionary Computation (CEC 2006)*, pp. 499–506. IEEE Press, New York (2006)
11. Fanelli, R.: Further experimentation with hybrid immune inspired network intrusion detection. In: Hart, E., McEwan, C., Timmis, J., Hone, A. (eds.) ICARIS 2010. LNCS, vol. 6209, pp. 264–275. Springer, Heidelberg (2010)

12. Drozda, M., Schaust, S., Szczerbicka, H.: Immuno-inspired Knowledge Management for Ad Hoc Wireless Networks. In: Szczerbicki, E., Nguyen, N.T. (eds.), pp. 1–26. Springer, Heidelberg (2010)
13. Marti, S., Giuli, T.J., Lai, K., Baker, M.: Mitigating routing misbehavior in mobile ad hoc networks. In: Proc. of International Conference on Mobile Computing and Networking, pp. 255–265 (2005)
14. Wang, D., Hu, M., Zhi, H.: A survey of Secure Routing in Ad Hoc Networks. In: 9th IEEE International Conference on Web Age Information Management, pp. 482–486. IEEE Press, Zhangjiajie Hunan (2008)
15. Perkins, C.E., Royer, E.M.: Ad hoc On-Demand Distance Vector Routing. In: Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, pp. 90–100 (1999)
16. Taneja, S., Kush, A.: A Survey of routing protocols in mobile ad hoc networks. *international journal of innovation management and technology* 1, 279–285 (2010)
17. Nadeem, A., Howarth, M.: Adaptive Intrusion Detection & Prevention of Denial of Service Attacks in MANETs. In: Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing, pp. 926–930. ACM, New York (2009)
18. United States Government. NIAIDS. Understanding the Immune System, How It Works. NIH Publication No. 03-5423. U.S. National Institutes of Health (2003)
19. University of Tokyo, Japan,
http://www.mcl.iis.u-tokyo.ac.jp/eng_version/index.html

Modeling, Analysis, and Characterization of Dubai Financial Market as a Social Network

Ahmed El Toukhy¹, Maytham Safar¹, and Khaled Mahdi²

¹Computer Engineering Department, Kuwait University

²Chemical Engineering Department, Kuwait University

ahmadtookhy@hotmail.com, maytham.safar@ku.edu.kw,
khaled.mahdi@ku.edu.kw

Abstract. A social network is a structure made up of nodes, which are also called social actors, linked together with edges, which are also called social links. Social networking has been evolving during the past years. A lot of researches have been conducted in that field and a lot of applications have been arisen. In social networking applications, a social network is constructed for the specific application field and then analyzed based on some parameters to try to understand, analyze, and predict the behavior of the constructed network. In this work, we analyze two social networks constructed for the Dubai Financial Market to try to understand and predict the behavior of the stock market as a result to the financial crises and the cause of the stock market collapse.

Keywords: Social Networks, Stock Market, Dubai Financial Market, Entropy, Cycles.

1 Introduction

The studies and researches conducted in the social networking field have been increasing and evolving in the past few years. A lot of applications have arisen to the field where social networking and analysis would be of great benefit to analyze, understand, and predict the behavior of the social networks constructed for the specific application area [3, 1]. Social networking analysis is a method of constructing a single or multiple social networks for a specific type of application consisting of nodes, edges, and information transferred and communicated between the different nodes via the different edges connecting them [9]. The application areas varies very widely from communication networks, information networks, sharing, financial, business, people, and many other types where different information are being transferred between the nodes. One of the social networks applications is stock markets. Stock markets affect the financial status and economy of the country. A stock market is a place where stocks, shares of listed companies, bonds, and other financial securities are being traded. The amount of money traded in the stock market is an indicator of how strong is the economy of the country and how big the volume of trading in its stock market can be an indicator of investments and business growth in this country [8].

However, stock markets are prone to collapses. During the last financial crises in 2008, many stock markets have collapsed. When a stock market collapses, the amount of trading in it would decrease drastically and the listed companies' shares would lose much of its value. Since stock markets and the amount of trade in it is an indicator of the economy strength of the country, when financial crises occurs and the shares values fall down, the stock market would collapse and so the economy of the country would lose its strength. This means losing many investments and business opportunities in the respective country meaning that the economy would suffer more and more. Using social networking analysis methods would be very useful because it would allow us to analyze the behavior of the stock market in terms of social networks parameters. Analyzing the stock market and identifying the main properties of it represented in a social network would enable us to understand and know the reasons behind such activities and reactions happening in the stock market as well as predicting the behavior of the market in case of collapse or even normal situation. Predicting the behavior of the stock market in different financial situations would allow the authorized people to take the suitable actions to prevent a major collapse to occur or increase the trading amount in the market and gaining more strength to the economy.

Dubai Financial Market (DFM) is the stock market of Dubai city in United Arab Emirates (UAE) where different types of securities are being traded like shares of listed companies and bonds [11]. It is one of three stock markets in UAE. DFM started its operation in 2000 with most of the listed companies based in UAE and few of them only are based in the different gulf countries [13]. DFM was found to invest in different financial securities to serve the economy of UAE with maximum safety, transparency, and fairness in the trading process. In this work, two networks are constructed from DFM. One is called Personnel Network where the nodes are the listed companies and an edge between two nodes (companies) represents one or more persons being board members in the boards of both companies. The other network is called Families Network where the nodes are the listed companies and an edge between two nodes represents one or more board members from each of the boards of the two companies belong to the same family. In order to construct those networks, we had to go to the website of each listed company in DFM or search for the company's latest yearly report to get a list of all its board members with their family names to know the common board members between the companies and identifying the members with the same family name from boards of different companies to identify the edges between the nodes in the two networks.

In order to analyze the two constructed networks, two methods are used. One is the cyclic analysis method where the number of cycles versus cycle degree (number of nodes in the cycle) is computed for each possible cycle degree and the Entropy of cycles is calculated for the network. For this method, two different algorithms are being used for the two networks. For the Personnel Network, an exact algorithm calculating the exact number of cycles for each cycle degree is used. In the Families Network, an approximation algorithm is being used because there are many edges between the nodes which makes the use of the exact algorithm not feasible and will take very long time. The other analysis method is non-cyclic where some parameters are computed to give a picture about some of the properties of the network and its behavior.

In the next sections, we will provide a background about stock markets in general, cycles and parameters calculation for social networks, and an explanation for the entropy of cycles and its calculation. After that, information about Dubai Financial Market, its history and properties, and the networks constructed and method of modeling them is presented. We then present the experimental work and results conducted on the two DFM networks including the entropy and parameters calculation. Finally, possible future expansion to our work as well as the conclusion of it is mentioned at the end.

2 Background and Related Work

In the coming subsections, we introduce stock markets, their importance, and some of their characteristics. We also introduce cycles as the modeling method used to model the social networks constructed in this work. Finally, we give an introduction about the non-cyclic analysis and parameters calculation conducted on the social networks.

2.1 Stock Market

Economy is becoming the moving force in nowadays world. It is considered a very important reason and motivation for many events and things happening in the world. Many wars have started because of economic and financial goals while many researches have been conducted only for pure economic motivations. Stock markets have always had huge effect on a country's economic status. It is actually considered the most important indicator for the status of a country's economy in terms of strength and stability. The stock market is a public market in which the shares of specific listed companies and firms are being exchanged (bought and sold) at an announced known price. The stocks exchanged are considered financial securities for those listed companies and organizations [16].

The stock market in any country is of great importance. It is a primary source of money for companies [16]. A company needing cash would sell some of its ownership shares in the stock market in order to expand its capital to get the needed amount and solve any financials issues it could face without the need to get a loan from a bank. The price of shares has a great impact on the company itself, and the economic activity in the whole market as well. High share prices tend to be related to the value of the company and the amount of business being done by it.

On the other hand, stock markets are always prone to collapses. A collapse would occur to the stock market in case of a sharp sudden change in shares price of listed securities. The stock market collapse can be a result of many various financial and economic factors. However, it can be said that the general reason for any stock market collapse is the lack of confidence in the country's economy [8]. This even can be a result of any political, public, or local issues. There have been famous stock market collapses varying in size and effect on the country and the whole world. The last collapse was the stock market collapse in 2008 which has affected most of the countries all over the world and is still affecting the economy of many countries until today. In this work, we analyze the Dubai Financial Market by constructing two networks from it and use the cyclic and non-cyclic parameters for analysis.

2.2 Cycles

The first part of network analysis conducted in this work is cyclic analysis and entropy calculation. In a social network, the state of the network can be defined according to several choices. One of these choices is the degree, defined as the number of links going in or out of an actor in the network. This definition is commonly used by most of the researchers. Different non-universal forms of distribution could result when analyzing the social network using the degree. For example, random network has a Poisson distribution of the degree. Small-World network has generalized binomial distribution [1]. There is no unified form reported. A different unified distribution form that is applicable for all social networks can be generated by using a different definition of the state of the network. The state is defined as the number of cycles existing in the network. Then we find the distribution function of the cycles and calculate the Entropy of the network [1].

2.3 Parameters

Beside the previous cyclic method of social networks analysis (Entropy calculation), there are other non-cyclic parameters that can be calculated to analyze the network further more. Those parameters include Degree Centrality, Betweenness Centrality, Closeness Centrality, Eccentricity, Eigenvector Centrality, Graph Diameter, Graph Density, and Clustering-Coefficient. Degree Centrality simply means the number of edges linked to a node. In directed graphs, two measures are considered, indegree and outdegree. Indegree refers to the edges pointing to a node while outdegree refers to the edges pointing out from the node [14]. Betweenness Centrality is a measure that describes the probability a node would arise in shortest paths between other nodes in the network. Closeness Centrality calculates the average number of nodes between a specific node and all other nodes in the network. Eccentricity refers to the distance between a specific given node and the farthest node from it in the network. Eigenvector Centrality is a measure corresponding to the node's importance in the graph based on the node's connections. This is done by assigning a score for each edge linked to each node so high-score edges would contribute to a node's score more than low-score edges [14]. Graph Diameter means the farthest distance between the two farthest nodes from each other in the graph while Graph Density calculates the closeness of the network to completeness. A complete network would have all possible edges between all nodes in it and so its density is equal to 1. Finally, the clustering coefficient is a measure that indicates how close a network to "small world". It indicates how nodes in the graph tend to cluster together. It is calculated for each node by itself as well as the average clustering coefficient for the whole graph which is the average of local clustering coefficient values across all nodes in the graph. A "small world" network has higher average clustering coefficient than a random diverse network [15].

3 Entropy and Cycles

Entropy is one of the most important analysis methods and indicator of social network behavior. One definition of Entropy is the degree of robustness of the network [4].

Giving this definition, it is assumed that the network is fully dynamic, meaning that the links can change without any constraint on their behavior. Therefore, Entropy can be defined as the level of disorder of a defined system. It can be understood from the definition that a dynamic changing network is of low entropy while a static rigid strongly connected network would have high Entropy. From a statistical point of view, Entropy means the probability $P(k)$ that the system is in a specific state k . It can be represented by the following equation:

$$S = - \sum_k P(k) \ln P(k) \quad (1)$$

The definition of the state of the system differs depending on the system itself. In biological systems, the state means the positions of the molecules while edges mean the interaction between them [1]. In social networks, the state of the system can be interpreted by several ways. One way is the degree, which is the number of links associated with each social actor (node) in the social network. This interpretation is commonly used by almost all the researchers. It is very easy to calculate; however, not accurate because it would result in different forms of distribution and so different analysis of the social networks [1].

Another interpretation of the state of a social network is cycles. Cycles are considered one of the major concerns and analysis methods in social networks. It results to a universal form of distribution even for different types of social networks. Using cycles to analyze the networks means in statistical mechanisms to find the probability that a social actor in the network would receive the information he sent again from one of the social actors linked to it [1]. In other words, if a network is non-cyclic, the entropy would be infinity because the network is very static and there is no way a social actor would receive the information it sent again.

In this work, we use the concept of entropy of cycles to analyze the social network of a stock market exchange and try to understand the behavior of such network and the relation between the performance of different listed companies in the stock market and the performance of the exchange market in total.

4 Dubai Financial Market

The Dubai Financial Market (DFM) is the stock exchange market based in the Trade Centre Building in Dubai city and started operations on 26 March 2000. It was founded to provide trading shares of Dubai, UAE, and some regional companies or Public Joint Stock Companies (PJSC) [13]. It is one of three stock exchange markets in United Arab Emirates (UAE) which are Dubai Financial Market (DFM), Dubai International Financial Exchange (DIFX), and Abu Dhabi Securities Exchange (ADX) [13]. DFM has currently 79 listed companies most of them based in UAE and categorized into 10 categories: Banks, Investments and Financial Services, Insurance, Real estate and Construction, Transportation, Materials, Consumer Staples, Telecommunication, Utilities, and NASDAQ Dubai. NASDAQ Dubai is formally called Dubai International Financial Market and it is a stock exchange market

founded in Dubai in 2005 aiming to be the main gateway of opportunities and stock exchange in the Gulf region, Middle East, South Africa, Turkey, as well as Central and South Asia [12].

DFM is considered the secondary market (primary market is DIFX) for trading of different types of securities. These types include public joint stock companies' shares, bonds issued by the federal government, or any of the local governments or public institutions of UAE, investment funds, or other types of financial instruments approved by the DFM [11].

Each stock exchange has a regularity authority. Dubai Financial Market's regulatory authority is Emirates Securities and Commodities Authority (ESCA). It is also the regulator for Abu Dhabi stock market exchange (ADX). However, a separate authority is responsible for the regulatory of DIFX which is Dubai Financial Services Authority (DFSA) [13].

DFM's mission is to "create a fair, efficient, liquid and transparent marketplace that provides choices through the best utilization of available resources in order to serve all stakeholders" [11]. As part of this mission, it was decided by an Executive Council Decree that DFM itself is a public joint stock company in the UAE with a capital of AED 8 Billion allocated over 8 Billion shares with value of AED 1 per share and 20% of the DFM shares are offered for public subscription.

The main objective of DFM is to invest in securities to serve the national economy of UAE and regulate the trading process in order to ensure maximum protection and safety achieved [11]. DFM helps creating and providing liquidity in the market by the fair trading practices between the investors as well as arranging the change and transfer of securities ownership through settlement and clearing mechanisms managed by an electronic system to guarantee maximum efficiency achieved during this process. Brokers present a very remarkable part and play a very important role in any stock exchange market and so handling them and the interaction between them and the stock market and the investors is one of the most important roles of a stock market. DFM implements certain rules and regulations for professional communication between the brokers and DFM staff to guarantee a high level of integrity. Reports are also a key factor because they show the status of the market and the securities listed in it which in turn helps the decision makers take better decisions on time. This can be achieved by collecting data and statistics and issuing reports on time and so the forecasting process can be done on time as well which is being achieved in DFM. [11].

5 Network Modeling and Characterization

In order to construct a social network for Dubai Financial Market, all the board members of all the listed companies have been gathered in one list by visiting all the websites of those companies. Then, two social networks are constructed for Dubai Financial Market. In the first one, named "Personnel Network", the companies are nodes (social actors), while an edge represents a person being a board member in the two companies connected by that edge. In other words, if a person is a board member in 4 companies, this means there is a link between each company and the other 3 companies.

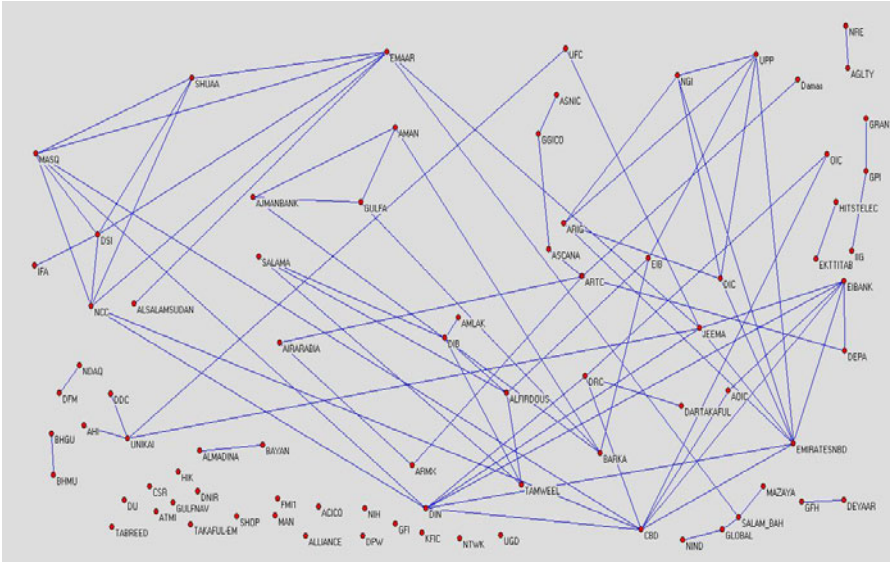


Fig. 1. Personnel Network of Dubai Financial Market

In the second network, named “Families Network”, the companies are nodes, same as the first one, while a link between two companies means that there are two board members from the same family in those two companies.

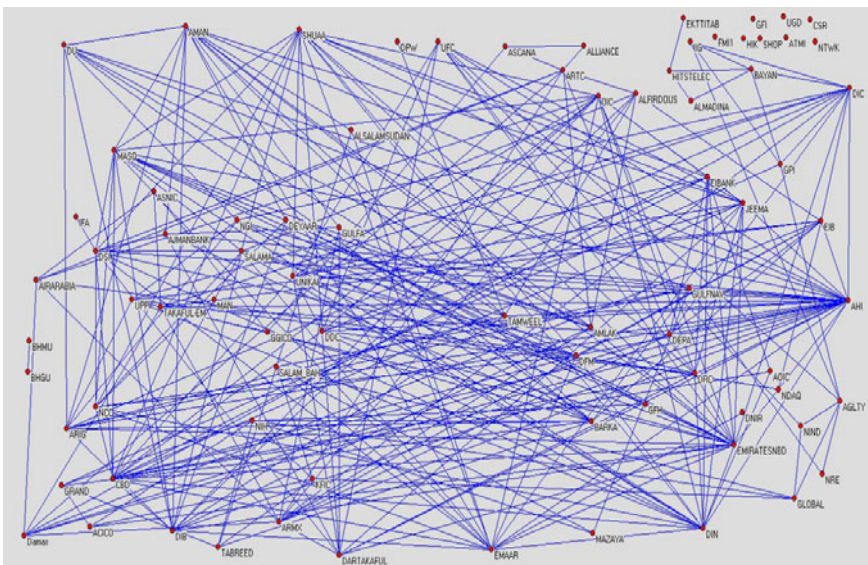


Fig. 2. Families Network of Dubai Financial Market

Logically, this network would be much more connected than the first one. This has led us to use the exact algorithm in order to calculate the Entropy of Cycles for the first network (Personnel Network), while use the approximation algorithm to calculate the probability of cycles in the second network (Families Network). The exact algorithm of computing cycles of a graph is NP-complete [2, 5]. Hence, applying it to highly connected large networks is not practical.

Both networks were constructed with undirected links. That is, if two companies have the same board member (or two board members with the same family), an edge is placed between the two networks, with no direction (the edge is considered to be in both directions). On the other hand, edges weight (cost) was not considered. If two companies have more than one board member in common between them, a link of regular cost (no additional weight is assigned) would be placed between them.

Pajek 2.0 network analysis software was used to draw the constructed networks while Gephi 0.7 network analysis software was used to calculate the non-cyclic parameters. The cyclic analysis (Entropy of cycles) was computed using both exact algorithm [3] for Personnel Network and approximation algorithm [5] for Families Network.

6 Experimental Work and Results

The purpose of this work is to apply the social networking analysis methods (Entropy of Cycles as well as calculation of non-cyclic parameters) to analyze and understand the behavior of Stock Markets, applied on Dubai Financial Market in this work, and to find any relation between this behavior and the market collapse occurred to it during the last financial crises. We are trying to relate between the board members of the listed companies and how they were affected by the financial crises. In this work, we are finding the relation between the market collapses resulted from the financial crises in 2008, and persons being board members in multiple listed companies.

Looking at the constructed networks (Figure1 and Figure2), we can see that the nodes (companies) in the networks are very connected and related to each other by many links. This would prove and explains the massive market collapse that happened in Dubai Financial Market as a result to the financial crises of 2008. Simply, because if a company's value is getting lower in the stock market, it would in turn affect many other companies since it is connected to multiple companies by having common board members as shown and illustrated in the constructed networks, which will result in the collapse of many other companies as well as the collapse of the entire market.

6.1 Cycles and Entropy Calculation

We started the experiment by collecting data about the board members of each listed company in DFM. Most of the board members we found easily in the "About us" page of the company's website. Some of them we could find the board members in the yearly report uploaded on the company's website. After constructing the two networks; Personnel and Families Networks, we used the exact algorithm to calculate the number of cycles versus the cycle length for the Personnel Network while the approximation algorithm was used to calculate the probability of having cycles versus the different cycle lengths. This is because the first network is much simpler and has

Table 1. Network parameters calculation and comparison between the two networks

Social Network	a	b	c	Cyclic Entropy	Network Type
Personnel	0.195	17.297	2.847	2.186	Small-World
Families	0.119	109.928	7.071	2.645	Scale-Free

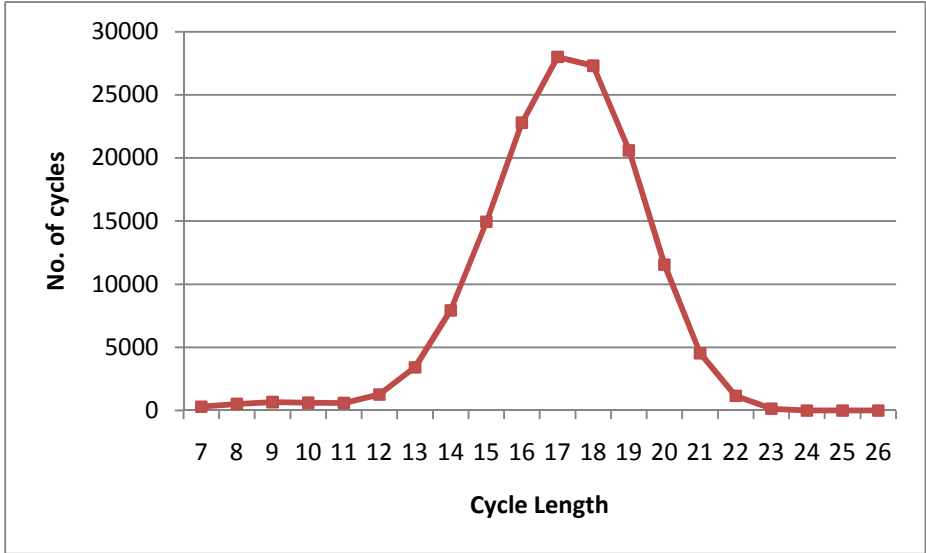


Fig. 3. Personnel Network cycles distribution vs. cycle length

less number of edges between its nodes than the second one which makes the exact algorithm run in a finite amount of time for the first network but not applicable for the second one. The Families network showed entropy of 2.65. The maximum cycle length along with the total number of cycles cannot be calculated for the Families Network because the approximation algorithm was used, not the exact. The Personnel network showed a maximum cycle length of 23 nodes, 146655 as total number of cycles of all lengths in the network, and entropy of 2.19.

Based on previous works by the authors [7], one technique to characterize the studied social networks is to evaluate the cyclic probability distributions and the corresponding cyclic entropy. Such a distribution has a universal class in the form of a Gaussian distribution $p_i = a \exp(-(x - b)^2 / c^2)$. It fits all types of social network. Each social network type has a reasonable difference range of values of the distribution constants a , b and c . Furthermore, the value of the network cyclic entropy is another property that can be used to identify the type of social network. In this study, we this method of social network characterization on the personnel and

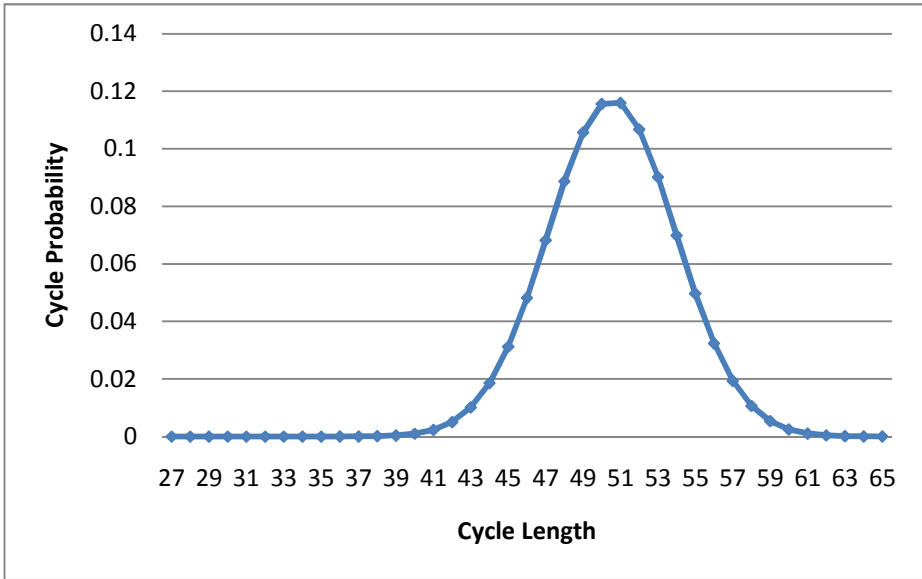


Fig. 4. Families Network cycles probabilities distribution vs. cycle length

families networks. Based on the values of distribution constants and the cyclic entropy, the personnel network is a form of a Small-World network and the families network is almost certain is a Scale-Free network. The results are summarized in Table 1. In other words, Dubai Financial Market on 2009 was families controlled and biased stock market as the types of social network indicate. A fair and unbiased market is expected to have a random social network structure. As suggested by [17], the lack of corporate governance in most Arab stock markets lead to a biased market prone to failure and collapse or on the other side well-protected but inflexible.

6.2 Degree-Based Parameters Calculation

Confirming the findings in section 6.1, further analyses to calculate known social network parameters in the literature for each network were done beside such parameters will provide another point of view of the social network behaviors. Some of the parameters are calculated for each node (company) in the network as a local value and then a network value is calculated based on the average. Some other parameters are calculated for the entire network directly. Some other parameters are related to edges in the network. All the parameters are calculated using Gephi 0.7 Beta software and the setup of the two constructed networks is undirected with no edge weight included. Table 2 compares the degree-based parameters of personnel and families networks.

Table 2. Network parameters calculation and comparison between the two networks

Parameter		Personnel Network	Families Network	
Degree	Highest value	8	28	
	Lowest value	0	0	
	Avg. Degree	2.125	7.2	
Graph Distance	Diameter	11	6	
	Avg. Path Length	4.5402	2.6012	
	No. of shortest paths	2088	4694	
Centrality	Betweenness	Highest value	436.92	392.1
		Lowest value	0	0
		Average value	46.2	46.975
	Closeness	Highest value	7.2	4.75
		Lowest value	0	0
		Average value	2.83	2.269
	Eccentricity	Highest value	11	6
		Lowest value	0	0
		Average value	4.91	4.2875
Density		0.0269	0.0911	
Clustering Coefficient	Avg. Clustering Coefficient	0.2574	0.4017	
	Total Triangles	44	515	
Eigenvector Centralities	Highest value	1	1	
	Lowest value	0	0	
	Average value	0.161	0.231	

7 Conclusion

In this work, we studied the relationship between the different listed companies in Dubai Financial Market (DFM) and constructed two networks out of it, the first one we called the Personnel Network while the other one we called the Families Network. In both networks, we assumed the companies are social actors while board members would form social links between nodes. We calculated exact number of cycles in the first network and approximated the probability of cycles in the second one. Then we computed the entropy of cycles for both networks. The entropy of the Families Network was larger than the entropy of the Personnel Network which means that the Families Network is more rigid. On the other hand, we calculated some non-cyclic parameters to understand the behavior and nature of the networks. From the figures showing the networks constructed, it is very obvious that the companies in DFM are very connected and linked to each other having common members in their boards or many members from the same family. Note that the links associated in the constructed networks have no weight. Some companies had more than one common member in their boards or more than two members from the same family. Another expansion for this work would be to give weight to the edges and reconstruct the networks to recalculate the entropy with more precise results. The analysis conducted in this work proved the huge market collapse occurred during the last financial crises in 2008 for

Dubai Financial Market, the stock market of Dubai city has a reason that is the lack of corporate governance due to the Small-World structure of the personnel network and the Scale-Free structure of the families network in the market.

8 Future Work

This work can be extended in the future by collecting the data regarding the listed companies and the board members of all those companies in Dubai Financial Market for the past 5 years and then construct the same two networks of listed companies as social actors and board members as social links. In this case, we would be able to determine the effect of the social network and relation between boards of listed companies on the market in Dubai and then would know how such relation between the boards has resulted in the bad stock market collapse in DFM by comparing the constructed networks and the change occurring in the stock market over time through the past 5 years.

References

1. Mahdi, K., Jammal, L., Safar, M.: Characterizing Collaborative Social Networks Using Cyclic Entropy, Case Study: Wikipedia. In: IADIS International Conference on Web Based Communities, pp. 125–130. Inderscience Publishers, Algarve (2009)
2. Safar, M., Farahat, H., Mahdi, K.: Analysis of Dynamic Social Network: E-mail Messages Exchange Network. In: 11th International Conference on Information Integration and Web-based Applications & Services (iiWAS), pp. 41–48. ACM, Kuala Lumpur Malaysia (2009)
3. Mahdi, K., Safar, M., Farahat, H.: Analysis of Temporal Evolution of Social Networks. *J. Mobile Multimedia* 5(4), 333–350 (2009)
4. Mahdi, K., Safar, M., Sorkhoh, I.: Entropy of Robust Social Networks. In: IADIS International e-Society Conference, iadis, Algarve, Portugal (2008)
5. Safar, M., Farahat, H., Kassem, A.: Approximate Cycles Count in Undirected Graphs. *J. DIM*
6. Mahdi, K., Safar, M., Sorkhoh, I., Kassem, A.: Cycle-Based versus Degree-based Classification of Social Networks. *J. DIM* 7(6), 383–389 (2009)
7. Safar, M., Mahdi, K., Kassem, A.: Universal Cycles Distribution Function of Social Networks. In: First International Conference on Networked Digital Technologies, pp. 354–359. IEEE Xplore, Ostrava (2009)
8. Stock Market, http://en.wikipedia.org/wiki/Stock_market
9. INSNA – What is Social Network Analysis, <http://www.insna.org/sna/what.html>
10. Dubai Financial Market, http://en.wikipedia.org/wiki/Dubai_Financial_Market
11. About, DFM Overview, <http://www.dfm.ae/pages/default.aspx?c=801>
12. NASDAQ Dubai, http://en.wikipedia.org/wiki/NASDAQ_Dubai
13. Dubai Financial Market (DFM), <http://www.sharewadi.com/dubai-financial-market.php>
14. Centrality, <http://en.wikipedia.org/wiki/Centrality>
15. Clustering Coefficient, http://en.wikipedia.org/wiki/Clustering_coefficient
16. Online Stock Trading Info, <http://www.onlinestocktradinginfo.com/>
17. Almajid, A., Riquelme, H., Safar, M., Mahdi, K.: Corporate Interlock Directorates in Kuwait Stock Exchange Market. Submitted to IADIS International Conference on Web Based Communities and Social Media, WBC (2011)

Secret-Eye: A Tool to Quantify User's Emotion and Discussion Issues through a Web-Based Forum

Siti Z.Z. Abidin, Nasiroh Omar, Muhammad H.M. Radzi,
and Mohammad B.C. Haron

Faculty of Computer and Mathematical Sciences
Universiti Teknologi MARA
Shah Alam, Selangor, MALAYSIA

Abstract. Receiving, synthesizing and communicating information to others are gradually more important. Information can be in many forms, and text can represent a lot of information that includes people's emotion. Emotion is usually measured through conducting a survey. For digital information, emotion can be expressed in words or emoticons. In this paper, we propose to quantify the user's perception in a web-based forum through analyzing words used by users. By determining the number of occurrences for each significant word in the forum, we measure the user's perception based on emotion and issues that they are discussing. Therefore, we implement a software tool called Secret-Eye that provides a platform to extract words in any web-based forum. The tool is implemented using C# language. It reads and filters the words before clustering them into emotion and facts. At the same time, all the significant words are counted according to the specified cluster. The tool also considers a few emoticons that are clustered as emotion. This approach is significant in identifying public opinion that can collect people's perception and discussion issues unobtrusively through case-based online forum. At this stage, the scope of this research is focused on a Malay language which will be extended to be used in other languages. The tool is important to Malaysian organizations, individuals and investigators who seek for public opinion as well as emotion through online forum. There are many tools for collecting public's perception, however, what is novel in this research is the use of case-based real time data collection method that highlights potential areas for using computer-based technology in quantifying public emotion and perception.

Keywords: Quantifying perception, Web-based, Real time data collection, Word, Emoticon, Emotion, Issues, Public opinion, Secret Eye.

1 Introduction

The use of internet allows ordinary people from all ages to express their opinions and feelings on any issue at their own convenience. People at young age are regularly expressed their views in complex sentences via e-questionnaires, forums or Weblogs. Moreover, they often choose to use natural expressions, specifically, in the form of "youth words" (Internet slang) and textual emoticons [1]. Although the people's

communication and comments are considered normal conversation and parts of modern social network, their perception may leads to threats [2], due to the fact that the perception of a person could be manipulated [3]. Originated from self-perception approach, it is possible to quantify people's opinion based on the texts that they use in their conversation. Self-perception is an awareness of the characteristic that constitute one's self. It means, people will decide on their own attitudes and feelings from watching themselves behave in various situations [4]. In addition, self perception is about looking a glass-self, how people think they may appear to others; how they think other people may evaluate their appearance and the resulting shame or pride others' feel [5]. In simpler words, self-perception can be defined as how people conduct themselves in different states and how other people evaluate about it.

Previously, the investigation on self-perception is carried out by using paper-based survey and people who participate are aware of the evaluation process and purpose of the survey. Since then, many tools for collecting public's self-perception have been advanced in parallel to the advancement of technology. Instead of conducting paper-based self-perception surveys, researchers can conduct computer-based self-perception survey. A review of literature shows that currently there is no existing system to quantify the perception and emotion of the received information via online media and produce instant feedback in a real-life setting.

This paper proposes a method to quantify people's perception in a web-based forum through analyzing words used by users. By determining the number of occurrences for each significant word in the forum, people's perception is quantified based on emotion and issues that they are discussing. Therefore, a software tool is implemented called *Secret-Eye* that provides a platform to extract words in any web-based forum. The tool is implemented using C# language. It reads and filters the words before clustering them into emotion and facts. At the same time, all the significant words are counted according to the specified cluster. The tool also considers a few emoticons that are clustered as emotion. This approach is significance in identifying public opinion that can collect people's perception and discussion issues unobtrusively through case-based online forum. At this stage, the scope of this research is focused on Malay language which will be extended to be used in other languages. The tool is important to Malaysian organizations, individuals and investigators who seek for public opinion as well as emotion in online forum. The novelty of this research is the use of real time data collection method that highlights potential areas for using computer-based technology in quantifying case-based public emotion and perception.

This paper is organized as follows. Section 2 discusses on research of quantifying people's perception. Next, Section 3 presents the framework and methodology of the software tool (*Secret-Eye*). Section 4 discusses the result and analysis of this research before the concluding remarks in Section 5.

2 Quantifying Perception

Perception is defined as the quality of understanding [6]. It includes the theory to understand the surrounding environment [7]. In Information Age, people usually

express their perception in digital words. The advancement of Internet technology allows people to express their perception along with emotion in many online platforms such as Weblog, online forum, chatting and emails. Emotion classification can identify the feelings of individuals toward specific events [8]. It is not a trivial task to extract emotional information from the lexical content or meaning of the words in a blog [9]. In addition to words, there are also the emotion icons (emoticons). The emoticons are widely used to represent emotional words [10]. Therefore, people's perception can be expressed by using simple but very meaningful icons which have the same meaning with the words that they want to use, such as quantifying emotion in a Weblog using a computerized system [9].

Methods in perception quantification have gradually changed from manual paper-based survey to automatic computer-based system. In paper-based survey, the preparation of questionnaire is tedious and time consuming. In addition, the choices of question and answer given to the participants are structured and limited to what is written in the likert-scale survey. Such limitation is not only due to the freedom of expressing opinion or emotion, but also in selecting the type and number of participants. Normally, the survey is sent to the chosen participants and they are expected to understand the questions and return back their responses within a specific time. Unfortunately, the response rate for paper-based survey is lower compared to internet-based survey [11]. In the paper-based survey, the participants' responses are collected for statistical analysis and visualization. These processes are also time consuming and its major drawback is its complexities require scientific knowledge and skill in statistics which may not be understood by layman audiences [12]. Thus, the computer-based survey provides the opportunities beyond paper-based capabilities in term of information structure, participants, and time to prepare, disseminate, collect and analyze participants' responses.

Currently, the Internet is increasingly popular for survey purposes and it is also used within the higher education environment for handling online information [13]. In education, the self-perception survey can be performed on students' learning to analyze teachers' Personal Style (PS) in the context of science and mathematics teaching [14]. With a new paradigm shift in analyzing information, the self-perception survey method can be transformed into quantification of people's perception. For example, instead of using self-perception survey in quantifying the level of reading comprehension, Omar *et al.* [15] propose a computer-based method. The method can quantify students' quantity of understanding in reading by using a task-based real time data collection in free-response and unstructured style. Similarly, in online media, people are free to express their perception of any issues with the same style. However, it is possible to include their perception embedded with the cues to express their emotion like in a face-to-face communication, which depends mostly on nonverbal reminders. Another way to express emotion is by using emoticons. The emoticons can provide nonverbal replacement, suggestive of facial expression, and may improve the exchange of emotional information by providing additional social cues beyond what is found in the verbal text of a message [16]. The emoticons can also be represented in words. Adapted from the online reading comprehension [15], this research uses similar quantification method to quantify unobtrusively people's perception and emotion through unstructured online information by using computer-based technology at real time and real life setting.

3 Secret-Eye Work Flow

In this research, a software tool for quantifying public’s perception called *Secret-Eye* is implemented. As a case study, a Malaysian online forum, namely *forum CARI*, a social media platform is used. This platform is meant for people to discuss any current issues. Anybody can sign as a member of the forum and participate at any time. People can hide their personalities by using nicknames and freely express their opinions. The reason of choosing this particular website is due to its popularity as the 15th most visited sites in Malaysia, the second most visited forum site and the first introduced forum in Malay language as the medium of conversation [17]. There are two phases in the implementation of the tool. The first phase is word extraction and the second phase is quantification process.

3.1 Extraction

Inspired by the Self-Perception Measurement Model [2], the extraction of forum’s content is performed by pulling out all the words. The flow of process during this phase is shown in Fig 1. The process starts when the forum’s website is found and the number of pages in the forum is determined. All words in the forum are read and stored in a text file for later processing. Therefore, any word that appears in the forum will be processed.

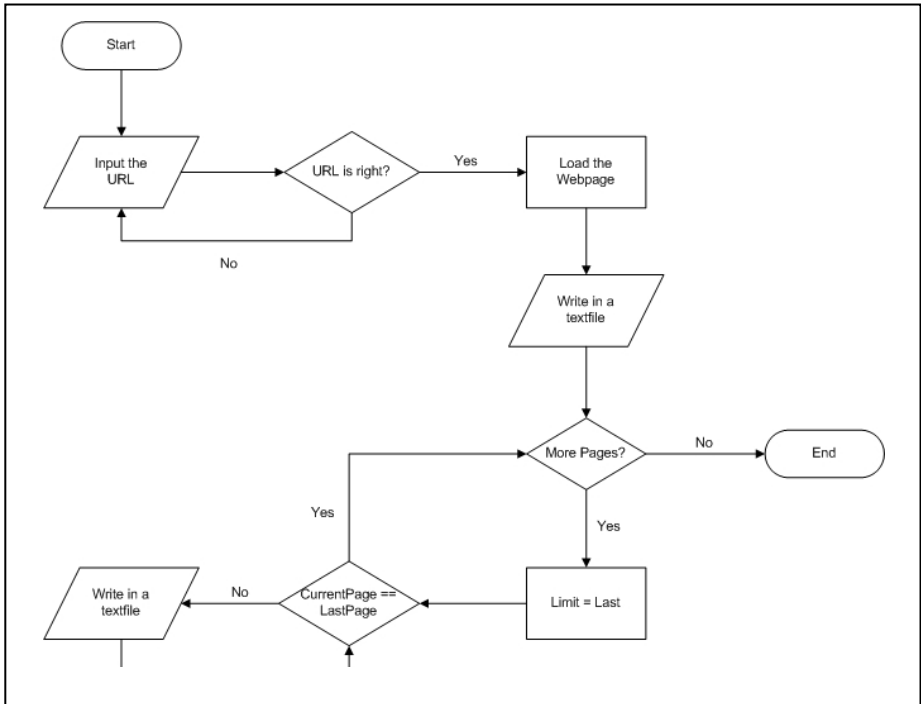


Fig. 1. Extraction Process

When the extraction process is finished, it is important to select the words that have significant meaning in the discussion and discard any word that acts as tag (in HTML file) or common word. All the significant words are saved as a plain text file. Based on this text file, the unique words and emoticons are manually selected and saved in a digital database, as part of case-based corpus for the language used in the forum.

3.2 Quantification

In the quantification process, the words and emoticons in the database are classified into *fact* and *emotion*. The process involves comparing each word in the text with words in the case-based corpus. Furthermore, the number of occurrences for each word and emoticon will be stored and ranked. The overall process involves evaluation, classification and calculation. The main purpose is to know how people feel during the discussion and the keywords that can portray the emotion and issues (based on *facts*) that people are discussing in the forum.

In order to validate the result on the quantification, a quick survey is manually conducted that involves 30 people who are selected at random from all ages regardless of their background. The majority of these people are of age 20 to 25. This survey will determine whether the suggested keywords are capable to give the clue about the discussion topic.

4 Results and Analysis

The design and implementation of *Secret-Eye* is in *C#* programming language. The tool provides a graphical user interface for the target user to use the system. The examples of target users are investigators or any individual who are interested in the public opinion on any case-based online forum. The results of this research are visualized in five graphical forms including bar charts and pie chart. Fig.2 displays the screenshot of *Secret-Eye* that presents the content of a text file with all the significant words after the tags and common words cleaning process, while Fig.3 shows the process of word search and classification in relation to the corpus in the database.

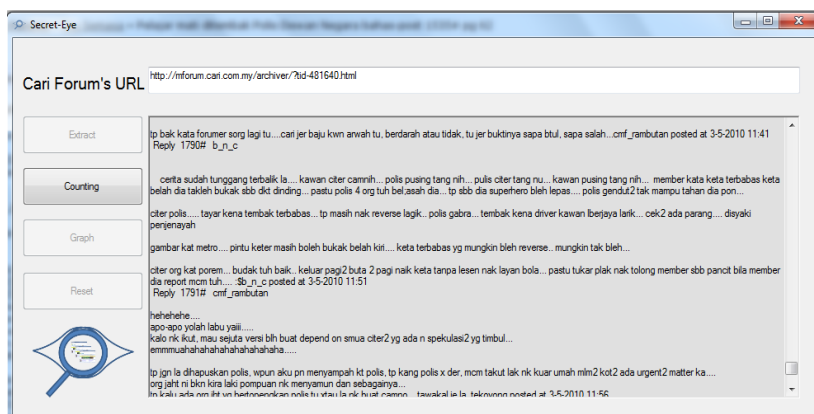


Fig. 2. Display of the filtered file

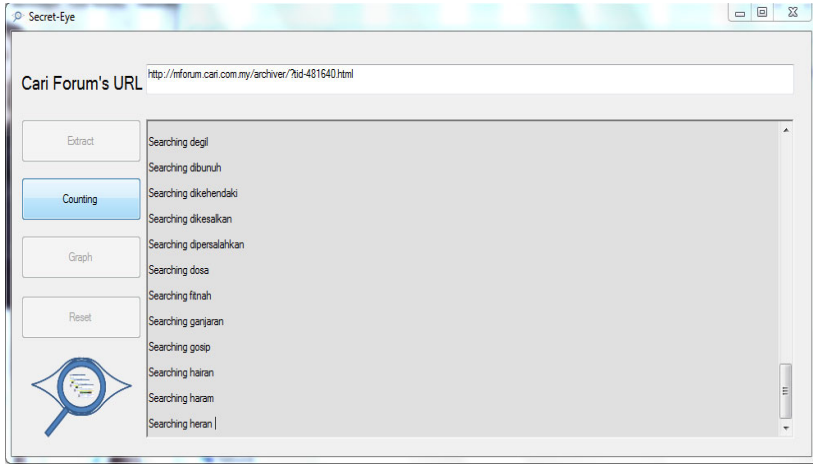


Fig. 3. The process of searching word occurrences in corpus

Fig.4 illustrates the output that contains the table of word occurrences after comparing the selected word with the case-based corpus dependent database. The result is presented in a bar chart of the top ten keywords for determining discussion issues.

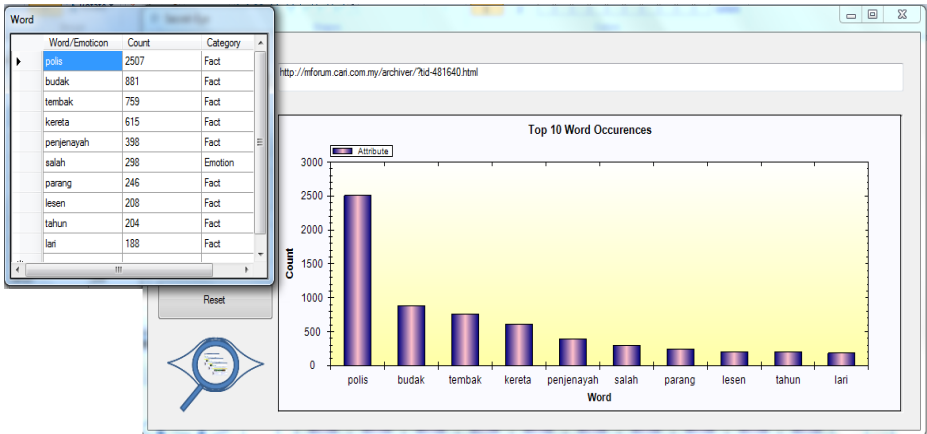


Fig. 4. The occurrences of top ten keywords on discussion issues

Fig.5 shows the two results of the quantification in a pie chart form: (a) top ten keywords for presenting people's emotion; and (b) the ratio of words for emotion in relative to facts. This ratio is calculated to check whether the discussion in the online forum is more towards emotion or facts. Therefore, this value can indicate whether most people are emotional or serious in the discussion.

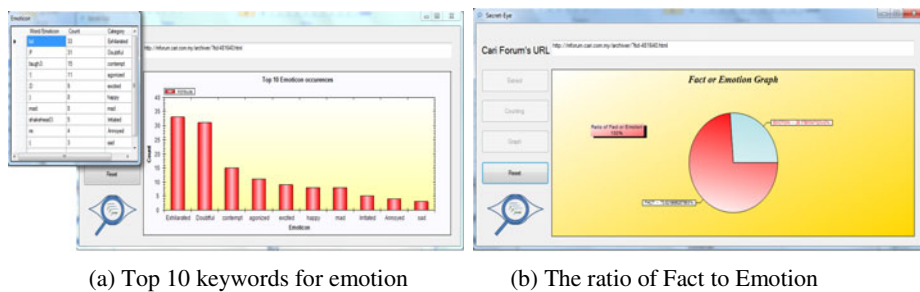


Fig. 5. Results of top ten keywords for emotion and the ratio of fact to emotion in the forum

The other two results present the bar chart of top ten keywords for emotion based on emoticons and top ten keywords on facts. Based on the quick paper-based survey with 32 participants, 28 of them are able to identify the discussion topic correctly when they refer to the graph of top ten keywords. The outcome of the survey indicates that the proposed method of quantifying virtual public's perception is feasible to capture the clue in the online forum discussion.

Since this work is originated from self perception measurement model [2] based on the web text handling model [15], this research is unique and one of its kind. Thus, the nearest comparable research work found in the literature is extracting keywords from online Thai texts and classify the keywords according to emotions [18]. Nevertheless, the research focuses on only emotion rather than issues and emoticons.

5 Conclusion

In this paper, we have presented the first potential attempt to quantify people's perception and their emotion of specific issues through capturing top ten keywords that are categorized into emotion and facts through case based online forum, performed at real time and real life situation. While most methods in quantifying people's perception are performed through survey-based methodology, we have shifted the paradigm by using a real-time data collection on a web-based case. By using the online forum, data collection on the public's opinion can be performed unobtrusively. At the same time, we argue that most people, who involve in the online discussion, will express their ideas freely and naturally.

The tool, named as *Secret-Eye* can extract all the words in the specified online forum and perform evaluation and analysis before producing four categories of top ten keywords; emotion, emoticon, facts, and occurrences. We conclude that it is feasible to capture the popular discussion topic that the public is chatting and arguing in the online forum. This new approach significantly reduce the time for collecting, processing, and analyzing data, as well as visualizing results, specifically for seeking public's perception and emotion of specific issues. It has the potential to be strengthened in helping any individual or organization who seeks public's opinion in making decision. In the future, the content of the database dictionary will be expanded to include many languages and issues.

Acknowledgements

We would like to thank the Universiti Teknologi MARA Malaysia for the financial support (Excellence Fund).

References

1. Itoh, M.: Contextual analysis of Complex Sentences Expressing Sentiments. In: 2009 Eight International Symposium on Natural Language Processing, pp. 5–10. IEEE, Los Alamitos (2009)
2. Omar, N., Abidin, S.Z.: Towards Measuring Self-Perception in Disseminating Information. In: Proceedings 2010 International Conference on Information Retrieval and Knowledge Management, CAMP 2010, Shah Alam, Malaysia, pp. 147–151 (2010)
3. Lawrence, D.: Measures of Effects. Information Operation Course. FKPM, UiTM, Malaysia (2009)
4. Bem, D.J.: Self-Perception: An Alternative Interpretation of Cognitive Dissonance Phenomena. *Psychological Review*, 183–200 (1967)
5. Self Perception,
http://webpace.ship.edu/ambart/PSY_220/Self_percol.htm
6. Hornby, A.S. (ed.): Perception. Oxford Fajar Dictionary, p. 1335. Shah Alam (2000)
7. Sun Libo, L.Y.: The Hierarchical Perception Model for Crowd Simulation. In: Sixth International Conference on Computer Graphics, Imaging and Visualization, pp. 106–111 (2009)
8. Yang, C., Lin, K.H.-Y., Chen, H.-H.: Emotion Classification Using Web Blog Corpora. In: IEEE/WIC/ACM International Conference on Web Intelligence (2007)
9. Li, J., Ren, F.: F. Ren Ren Emotion Recognition from Blog Articles. In: International Conference on Natural Language Processing and Knowledge Engineering, IEEE NLP-KE 2008. Beijing, pp. 1–8 (2008)
10. Thompson, P.A., Foulger, D.A.: Effects of Pictographs and Quoting on Flaming in Electronic Mail. In: *Computers in Human Behavior*, pp. 225–243 (1996)
11. Cobanoglu, C., Warde, B., Moreo, P.J.: A Comparison of Mail, Fax and Web-based Survey Methods. *International Journal of Market Research* (2001)
12. Harrison, C.: Postmodern Research and e-learning: Anatomy and Representation. *European Educational Research Journal*, 80–93 (2006)
13. Leu, D.J., Neag, J., Neag, M.: Expanding the Reading Literacy Framework of PISA 2009 to Include Online Reading Comprehension. In: A working paper commissioned by the PISA 2009 Reading Expert Group (2009)
14. Ben-Chaim, D., Zoller, U.: Self-Perception versus Students' Perception of Teachers' Personal Style in College Science and Mathematics Courses. *Research in Science Education*, 437–454 (2001)
15. Omar, N., Higgins, C., Harrison, C., Campo Millan, D.: Evaluating Real-time Online Research Data (RORD) and Verbatim Quotient Detection (VQD): Low Inference Tools to Monitor Outcomes of Unconstrained Authentic Internet Research. In: Evaluating Real-time Online Research Data (RORD) and Verbatim Quotient Detection (VQD), Netherland, pp. 502–506 (2006)
16. Derks, D., Bos, A.E., Grumbkow, J.V.: Emoticons in Computer-Mediated Communication: Social Motives and Social Context. *CyberPsychology & Behavior* (2008)
17. Top Sites in Malaysia, <http://www.alexa.com/topsites/countries/MY>
18. Inrak, P., Sinthupinyo, S.: Applying Latent Semantic Analysis to Classify Emotions in Thai Text. In: International Conference on Computer Engineering and Technology, pp. 450–454 (2010)

A Concurrent Coloured Petri Nets Model for Solving Binary Search Problem on a Multicore Architecture

Alaa M. Al-Obaidi and Sai Peck Lee

Faculty of Computer Science and Information Technology,
University of Malaya, 50603 Kuala Lumpur, Malaysia
alaa@siswa.um.edu.my, saipeck@um.edu.my

Abstract. Multicore technology has imposed new ways of thinking in the field of software designs. In general, softwares should be adopted to reflect the hardware changes in the design process. Binary Search has its share in these developments, in the sense that this technique should be adaptable with the new environment. In this study, we develop a new hierarchical concurrent model that utilizes concurrent multithreaded scheduling in performing Binary Search on a multicore architecture. A novel algorithm and an enhanced algorithm have been proposed to control the mechanism of the concurrent model. The hierarchical model provides a solution for several weakness points in the previous studies such as overflow, fair distribution, and design complexity. The model has been simulated and verified by using Coloured Petri Nets as a language of modelling and through utilizing CPN-Tool as the modelling tool. The desired results have been reached with no errors or ambiguity.

Keywords: Divide and Conquer, Multithreading Scheduling, Work Stealing, Concurrency, Binary Search.

1 Introduction

Binary Search (BS) algorithm is one of Divide and Conquer (D&C) [1] [2] algorithms that is dedicated for finding a certain element in an ordered array (list). The BS technique starts by comparing the input value with the element in the middle of the ordered array. If no matching exists then the result of comparison determines in which array's half the process will be repeated. In case that the input value is less than the element in the middle; then the algorithm will be replicated only on the elements that are come before the element in the middle (left half). Otherwise, the searching will be focused only on the elements that come after the element in the middle (right half). The algorithm which is considered as an algorithmic function has a running time of $O(\log N)$ and it can be applied iteratively or recursively. A major downside in this algorithm happens when new elements are added to the array. This will enforce to resort the array again prior to any new searching [3] [4].

In a single processor system, BS algorithm has to be executed serially. However, with the advent in multicore technology, the technique has to be changed. Software designers found that multithreading could be the perfect software solution to cope

with the increasing number of cores per chip by using concurrency [5]. Modelling concurrent multithreaded systems represents a great challenge due to the nondeterministic nature of such systems besides the difficulty in synchronizing the threads. Concurrency, multithreading, and D&C technique have a common relation. The main problem can be divided into several parts; each part can be assigned to a thread. The ability to allocate a core for each thread makes all the threads working concurrently.

In this study we present a new hierarchical model that is able to schedule the actions of the BS algorithm on a simulated multicore environment. The heart of the model works under two new algorithms: Binary Search Multithreading Scheduling (BSMS) which is designed to schedule threads actions inside each modelled core. The BSMS algorithm is responsible for thread creation, dividing, and computing. The second algorithm is an enhanced algorithm, Enhanced Work Stealing Scheduling, (EWSS) algorithm which works as a coordinator between the modelled cores. The EWSS algorithm controls threads movement between the modelled cores. The algorithm is in charge of making all the cores working concurrently through stealing threads from victim (non-empty or working) cores and send them to the thief (empty or idle) cores. We applied these algorithms by using Coloured Petri Net (CPN) [6] [7] [8] [9] as a language of modelling, Coloured Petri Nets Meta Language (CPN-ML) [10] as a language of coding and CPN-Tool [10] as a software tool which enables us to create, simulate and verify the correctness of the designed models.

The rest of this article is organized as follows: Section 2 represents a background of Work-Stealing scheduling algorithm besides some information about Coloured Petri Nets (CPN), CPN-Tool and CPN-ML. Section 3 is dedicated for the problem statement. In Section 4, we demonstrate the research methodology that we propose. The proposed concurrent multithreaded scheduling model for solving Binary Search is explained in Section 5. Section 6 is dedicated for building the CPN model. The results of simulation are included in Section 7. Sections 8 and 9 are devoted for the discussion and conclusion. Finally, an Appendix which comprises all the CPN-ML code has been added at the end.

2 Background

There are two types of scheduling algorithms in a multicore environment: Work Sharing and Work Stealing. In Work Sharing, the scheduler continuously attempts to transfer threads from the core that has generated them to other cores hopefully in a fair load distributing among the cores. In Work Stealing, on the other hand, the idle cores attempt to steal threads from the working cores. Movement of threads between cores happens less frequently in Work Stealing than in Work Sharing. In Work Stealing, each core is accompanied by a local memory which is organized as a deque. A core can push threads into the deque from one end and pop threads from the other end. When a core becomes idle (out of threads), it tries to steal thread(s) from another deque using the other end of that deque. In general, each core has two statuses: either it is a thief (trying to steal threads from other cores) or a victim (other cores trying to steal from it).

The work of Blumofe and Leiserson [11] can be considered as a landmark in work stealing scheduling. They presented an algorithm that is able to schedule fully-strict (well-structured) multithreaded computations. Although the algorithm did well in the area that needs static space partition, it faces problems in the modern environments that support multiprogramming. The reason behind this drawback is the supposition of the availability of a fixed set of processors to achieve computation. Another important achievement of Work Stealing is the contribution of Arora et al [12]. They designed a thread scheduler for a shared-memory multiprocessors environment. They improved the work of Blumofe and Leiserson [11] by considering non fully-strict multithreaded computations in addition they dealt with a multiprogrammed environment instead of a dedicated one. However the huge success it gained as best choice algorithm for load balance both in academic and industrial fields, the algorithm of Arora et al faced two problems: it introduced memory management problem; having n processes with m as total memory size, the algorithm can deal with m/n threads in the deque at the most. In addition, overflows can easily occur; this means that the size of the array (deque) must be adjusted to solve the overflow problem. There is no simple way to free additional memory and continue. An expensive overflow mechanism must be added to fix this situation [13].

The work of Arora et al [12] has been extended by several ways: Hendler and Shavit presented in [14] the idea of stealing the half. In their algorithm, the process can steal up to the half number of items in the deque. The data locality of Work Stealing has been studied by Acar et al [15]; they presented a locality-guided work stealing algorithm that improves the data locality of multithreaded computations by allowing a thread to have an affinity for a processor. A major development on the algorithm of Arora et al has been proposed by Hendler et al in [13]; their algorithm detects synchronization conflicts by pointer-crossing instead of gaps between indexes as in the algorithm of Arora et al. The algorithm builds non-blocking dynamic-sized work stealing deques. It eliminates the need for any kind of application for fixing the overflow problem as in the algorithm of Arora et al. However, since lists represent the main structure in this algorithm, there is a kind of trade-off between space and time complexity for the reason that the work needed to maintain lists. A simple lock-free work stealing deque has been presented by Chase and Lev [16]. They used a cyclic array to store the elements. The algorithm can easily deal with overflow due to the cyclic nature of its data structure. There are no limits in this algorithm other than memory and integer sizes. The algorithm is simple and does not need garbage collector.

Coloured Petri Nets [6] is a graphical discrete-event language designed to model and validates concurrent systems. In addition to concurrency, communication and synchronization between the elements of the nets have a significant role in controlling the execution of the model. CPN has been developed from Petri Nets [17] [18] as being the origin of CPN. The main difference between Petri Nets and CPN is the addition of types to CPN besides the ability to write expressions and functions written in Standard Meta Language (SML) [19][20]. CPN model is an executable model in the sense that the process of execution shows different states of the system that is represented by the model.

CPN-Tool [10] is a software tool developed by Kurt Jensen [6][7] at University of Arhus (Denmark). The tool provides all the necessary facilities to create, simulate and

validate Coloured Petri Nets. CPN-Tool is a GUI tool that provides all the interaction methods such as menus and toolbars besides giving feedback messages when errors have been encountered during the process of code's syntax checking. CPN-Tool uses Coloured Petri Nets Meta Language (CPN-ML) as a language of writing declarations, expressions and codes. CPN-ML [10] is a language for writing nets inscriptions which includes expressions on the arcs, codes that control transitions as well as the declarations of the types and variables that are included in the net. It has been built based on SML [19][20].

3 Problem Statement

Multicore technology opens the door for new developments in the hardware side. The absence of investing these developments in the software side will make no real difference when executing the same old algorithms on the multicore environment. A key factor to adapt the current softwares with new hardware is through designing a multithreaded scheduler that makes the entire cores working concurrently. Modelling concurrent multithreaded scheduling systems represents the first step towards building real systems. Modelling languages and tools will no doubt catch many errors in the designing phase besides it ensures the stability of the system. This study aims to design a concurrent multithreaded scheduler that can solve Binary Search problem on a modelled multicore environment.

4 Research Methodology

The general research plan that we propose for building a concurrent multithreaded scheduling model for Binary Search problem consists of two phases: High-Level Scheduling and Core Scheduling Phases. Fig. 1 shows these two phases and the relation between them.

4.1 High-Level Scheduling Phase

High-Level Scheduling Phase has the duty of controlling threads redistribution among the cores. The process of redistribution can be achieved through stealing the threads from a certain set of cores, called the victim cores, to another set of cores called the thief cores. Victim cores are those cores that have one or more threads while thief cores are those who have no threads. We have developed new strategies to enhance the way of stealing (moving). In this paper, we propose a new strategy, Single-Step Stealing Strategy. This strategy simply can be applied as follows: When one or more cores are idle and there is a chance to allow them to steal, then the victim core(s) permit all the thief cores to steal in one step. The stealing process happens in a clock wise. Assuming that all the cores are organized on a single line, each thief core will check the non idle (victim) core to its right, then the other next core, etc. In case only one victim is available, this victim core will be the target for the thief core. When all the cores are busy or idle, nothing happens. The Single-Step Stealing Strategy has been applied through a function written in CPN-ML which receives, as inputs, list of threads that belong to the modelled cores. The output of the function will be the

updating of these lists. A major achievement in this function is the ability to redistribute the threads in the entire cores in one step, further, the function works concurrently with the other activities of the cores. In other words, while the High-Level Scheduling Phase balances the threads among the cores, each core can execute its own Core Scheduling Phase.

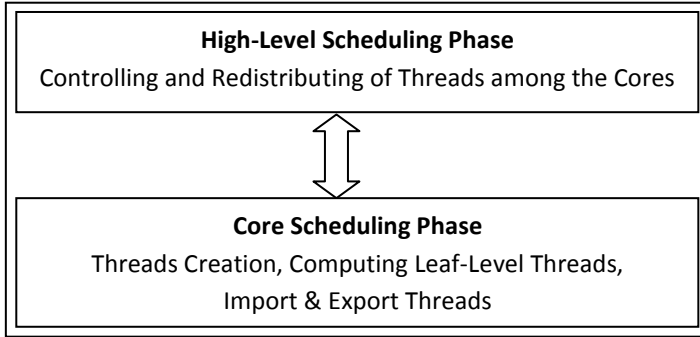


Fig. 1. The General Research Plan

4.2 Core Scheduling Phase

Core Scheduling Phase is responsible for scheduling operations inside each core besides facilitating the communication with the High-Level Scheduling Phase. A major significant in the Core Scheduling Phase is that all the cores work concurrently in scheduling their internal activities. The phase is in charge of performing three tasks: Threads creation, computing leaf-level threads and import / export threads. In fact, the process of threads division is done in this phase. The division process includes the creation of two threads; left and right. The left one has the same number as of its direct ancestor thread number multiplied by two. The right child holds the left child number plus one. A local memory organized as a stack (represented as a list of threads) is used to temporarily store the divided threads. The mechanism of this phase has been designed to create and PUSH/POP threads into the stack. In addition, the phase calculates leaf-level threads (leaf-level threads are those threads that are designed to be calculated directly). Finally, the Core Scheduling Phase cooperates with the High-Level Scheduling Phase in importing/exporting threads. Threads can be easily taken (stolen)/added from/to the stacks and redistributed to other cores resulting in a fair load distribution. The modelling tool, CPN-Tool, supports such cooperation through using Port-Socket mechanism [10].

5 A Proposed Concurrent Multithreaded Scheduling Model for Solving Binary Search Problem

In this study, we propose a concurrent model that is able to schedule Binary Search problem as multithreading scheduling. We can visualize the model as the one in Fig. 2. The model consists of two sub models namely: Coordinator Model and Core Model.

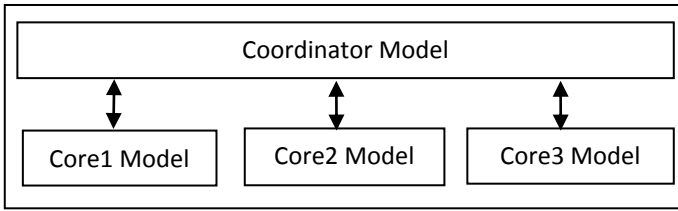


Fig. 2. The Binary Search Concurrent Multithreaded Scheduling Model

5.1 The Binary Search Coordinator Model

The Binary Search Coordinator Model represents the High-Level Scheduling Phase; it works under the control of a new algorithm, Enhanced Work Stealing Scheduling (EWSS), which represents the new development that we have achieved on work stealing algorithm. This model is responsible for coordinating threads moving between the cores in order to achieve load balance. The function EWSS as listed in the Appendix receives three parameters. Each parameter stands for a list of threads that belong to one core. The function EWSS redistributes the threads by moving threads from the victim cores to the thief cores. The Single-Step Stealing Strategy has been applied in this function. The EWSS function is built on the base of Pattern Matching technique which has the ability to redistribute the threads in one move (step). The statuses in bold (Appendix) are the only statuses which include threads redistribution. The rest of the statuses do not include redistribution either because there is no need or there is no possibility to steal (no thief cores).

5.2 The Binary Search Core Model

The Core Scheduling Phase is represented in the core model. The core model is controlled by a new algorithm; Binary Search Multithreading Scheduling (BSMS) that achieves threads creation and computing leaf-level threads. Every core schedules its activities by executing its own copy of BSMS function as listed in the Appendix resulting in a concurrent working of the entire cores. The thread is modelled as a 5-tuple: (ThreadId, ThreadFather, Element, Starting, Ending), where the first two parameters represent the thread’s number and the thread’s father number. ThreadId and FatherId are denoted as “Tx” (x is a positive integer number). The third parameter, Element, holds the input value that we are search for. The last two parameters (Starting and Ending) carry the first and last indices of the searching range (sorted list).

The BSMS function receives four parameters; List of Integers, List of Threads, Continuing Flag, and Location. The first parameter represents an ordered list of integers (searching area). The second parameter holds the list of threads. Third parameter is a Boolean variable which by default holds the value “false”. This value can be changed to “true” when the searching element has been found. The third parameter has the duty of stopping all the threads in continuing their searching process in case one of the threads succeeded in finding the desired element. Last parameter carries the location of searched element. By default the value of Location is

~ 1 (in CPN-ML, the negative symbol is \sim). In case the searching succeeded in finding the element then the value ~ 1 will be changed to hold the desired location otherwise it will remain ~ 1 .

The BSMS function uses a one let-in-end [10] construct in its body. The function pops the thread at the top of the stack (List of Threads), copies its parameters (ThreadId, ThreadFather, Element, Starting, and Ending) then it does one of the following choices:

A- If the Starting and Ending indices have the same value then the current thread is a leaf-level thread. The scheduler compares the value of the Element with value stored in the location indexed by Starting (or Ending since they have the same value). If we have a matching then the Continuing Flag gets the value “true” and Location gets the value of Starting. This choice is represented by the function leaf in the Appendix.

B- If the Starting is equal to (Ending + 1) then we have a semi leaf-level thread. A semi leaf-level thread is the ancestor of two leaf-level threads. The BSMS function compares the value of Element with values stored in the locations indexed by Starting and Starting + 1 (Ending). As in A, the values of Continuing Flag and Location alter according to the result of searching. The function semi_leaf (Appendix) represents this choice.

C- In case the (Starting + 1) is less than the Ending index then we calculate, a Middle index as: $Middle \leftarrow (Starting + Ending) / 2$. Now, a comparison is made first between the Element and the list's element indexed by Middle. If no matching happened, two new children threads are created: Left and Right threads. The left thread will have the following indices: Starting \leftarrow Same Starting, Ending \leftarrow Middle. The indices of the right child will be: Starting \leftarrow Middle + 1, Ending \leftarrow Same Ending. Finally, the two threads are pushed inside the stack. The Right_Child_Id and Left_Child_Id functions are used to create the left and right child. The difference between these two functions is that the first one calculates the left child number through multiplying ThreadId by two while the second function (Right_Child_Id) has the same number as the left child plus one. This choice is represented by the function others.

6 Building the CPN Binary Search Model Elements

The main page for the CPN Binary Search Model is shown in Fig. 3. This page includes three places (Core1List, Core2List, and Core3List), one transition (Coordinator), and three substituted transitions (Core1, Core2 and Core3). In CPN, a place is an oval shape which holds data. In our model, each place holds a list (stack) of threads. The initial value of the place is located at the top of each place. Place Core1List contains a list with a single thread, [(“T1”, “T0”, 87, 0, 39)] as an example, where “T1” and “T0” symbolize ThreadId and Thread's father Id respectively. The number 87 represents the Element parameter (for example, we are searching for the value 87). The zero stands for the Starting index and the number 39 stands for the Ending index (i.e. we have an ordered list with 40 elements). The initial value of Core2 and Core3 is a nil list ([]) since the execution starts with one main thread located in Core1. Each core has also a current value. This value will be changed during the execution of the model. The current values of the cores are located below the places. In general, a current value indicated by a circle (holds the number of tokens) and a rectangle (tokens details).

In CPN, transitions represent the action units. Transition Coordinator is in charge of executing the High-Level Scheduling Phase (EWSS function). The transition reads the lists of the cores through (C1In, C2In, and C3In), update the lists, and then send the feedback as C1Out, C2Out, and C3Out. To activate the transition Coordinator, the Guardian function as listed in the Appendix should return a Boolean True value. The function returns true only if there is at least one victim and at least one thief. In Fig. 3, the code above the Guardian function represents the code that will be carried out when transition Coordinator is executed by the CPN-Tool’s simulator. As illustrated in the figure, the transition executes the EWSS function which results in a fair distribution of the threads. Below each place there is a substituted transition: Core1, Core2, and Core3. Each substituted transition corresponds to a page similar to the one in Fig. 4 which represents Core1.

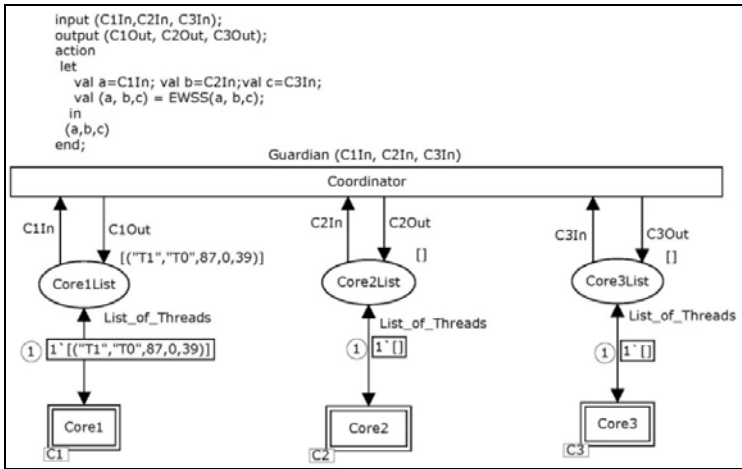


Fig. 3. A CPN Binary Search Model (Main Page)

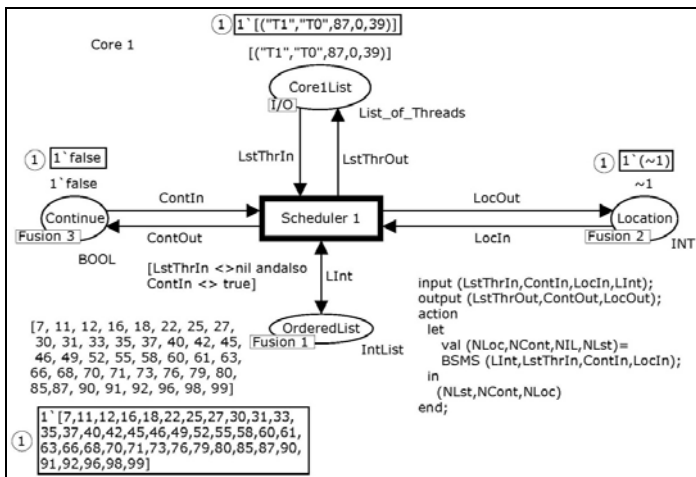


Fig. 4. A CPN Binary Search Core Model

The page in Fig. 4 shows the content of Core1. The core consists of four places; Core1List, OrderedList, Continue, and Location. In addition to one transition Scheduler 1. Transition Scheduler 1 receives four inputs:

1- LstThrIn (List of threads): The Scheduler 1 transition reads a list of threads (stored in the Place Core1List) which is represented by LstThrInt, updates it and sends it back as LstThrOut. The I/O symbol at the lower left corner of the place Core1List represents a port tag which is a kind of mechanism offered by the CPN-Tool to connect places from different pages. It allows the Coordinator to add/take threads to/from the place. This means that the place Core1List in Fig. 3 is just a copy of the place Core1List in Fig. 4. This kind of hierarchic simplifies the communication between the pages' elements in the model.

2- LInt (List of an ordered integer numbers): The transition also reads a list of ordered integers represented by LInt (stored in the Place OrderedList), however, the transition never changes the list (this why we have a bidirectional arrow). The Place OrderedList has an initial value which consists of an ordered integer list (40 elements). The initial value is located at the lower left corner of the Place OrderedList and the current value is located below the initial value. The Place Core1List has a Fusion 1 tag. Fused places [10] represent one of the CPN-Tool mechanisms in creating shared areas. In our example, the Place OrderdList in each core will share the same integer list.

3- LocIn (Location of the Element that we search for): The LocIn input represents an integer value (stored in the Place Location) which by default holds the value ~ 1 (minus 1). The value is changed (through LocOut) to hold the location of the element in case the element has been found otherwise it remains ~ 1 . The Place Location has a Fused tag, i.e. all the Location places in the entire cores share the same value.

4- ContIn (Continuous Boolean flag): Finally, the ContIn input stands for a Boolean value which is by default is "false" stored in the Place Continue. This value is changed to "true" through ContOut if the searched Element has been found. The "true" value is important to stop other threads from continuing their searching in case the Element has been found. As in the Place Location, all the Continue places are fused with Fusion 3 tag.

The Continue transition has a guard expression ($([LstThrIn \langle \rangle \text{nil andalso ContIn} \langle \rangle \text{true}]$) located at the lower left corner of the transition. This Boolean expression controls the activation of the transition. The expression returns true (allows the transition to run) only if the list of threads is not empty and (in CPN-ML, andalso stands for logical AND) ContIn is false, i.e. no threads succeeded in finding the desired Element.

7 The Results of Simulation

CPN-Tool is a single thread tool, it chooses transitions randomly. At every stage, the tool checks the active transitions (those who have enough tokens in their input places to enable their transition, beside that the guards, if exist, return true). The following represents an example of a simulation process that we have achieved on the model.

The main thread is [(“T1”,”T0”,87,0,39)] located in Core1. We want to search for the value 87 in the ordered list which is stored in Place OrderedList. Initially all the places Core2list and Core3List are empty. The simulation process starts by executing the only active transition, i.e. Scheduler 1 in Core number 1. Table 1 listed all the transitions activations.

Table 1. The Results of Simulation

No	Selected Transition	Core 1 List of Threads	Core 2 List of Threads	Core 3 List of Threads
1	Scheduler 1	(T2,T1,87,0,18), (T3,T1,87,20,39)	Nil	Nil
2	Coordinator	Nil	(T2,T1,87,0,18)	(T3,T1,87,20,39)
3	Scheduler 3	Nil	(T2,T1,87,0,18)	(T6,T3,87,20,28), (T7,T3,87,30,39)
4	Coordinator	(T6,T3,87,20,28)	(T2,T1,87,0,18)	(T7,T3,87,30,39)
5	Scheduler 2	(T6,T3,87,20,28)	(T4,T2,87,0,8), (T5,T2,87,10,18)	(T7,T3,87,30,39)
6	Scheduler 1	(T12,T6,87,20,23), (T13,T6,87,25,28)	(T4,T2,87,0,8), (T5,T2,87,10,18)	(T7,T3,87,30,39)
7	Scheduler 2	(T12,T6,87,20,23), (T13,T6,87,25,28)	(T8,T4,87,0,3), (T9,T4,87,5,8), (T5,T2,87,10,18)	(T7,T3,87,30,39)
8	Scheduler 1	(T24,T12,87,20,20), (T25,T12,87,22,23), (T13,T6,87,25,28)	(T8,T4,87,0,3), (T9,T4,87,5,8), (T5,T2,87,10,18)	(T7,T3,87,30,39)
9	Scheduler 2	(T24,T12,87,20,20), (T25,T12,87,22,23), (T13,T6,87,25,28)	(T16,T8,87,0,0), (T17,T8,87,2,3), (T9,T4,87,5,8), (T5,T2,87,10,18)	(T7,T3,87,30,39)
10	Scheduler 1	(T25,T12,87,22,23), (T13,T6,87,25,28)	(T16,T8,87,0,0), (T17,T8,87,2,3), (T9,T4,87,5,8), (T5,T2,87,10,18)	(T7,T3,87,30,39)
11	Scheduler 1	(T13,T6,87,25,28)	(T16,T8,87,0,0), (T17,T8,87,2,3), (T9,T4,87,5,8), (T5,T2,87,10,18)	(T7,T3,87,30,39)
12	Scheduler 2	(T13,T6,87,25,28)	(T17,T8,87,2,3), (T9,T4,87,5,8), (T5,T2,87,10,18)	(T7,T3,87,30,39)
13	Scheduler 3	(T13,T6,87,25,28)	(T17,T8,87,2,3), (T9,T4,87,5,8), (T5,T2,87,10,18)	(T14,T7,87,30,33), (T15,T7,87,35,39)
14	Scheduler 2	(T13,T6,87,25,28)	(T9,T4,87,5,8), (T5,T2,87,10,18)	(T14,T7,87,30,33), (T15,T7,87,35,39)
15	Scheduler 1	(T26,T13,87,25,25), (T27,T13,87,27,28)	(T9,T4,87,5,8), (T5,T2,87,10,18)	(T14,T7,87,30,33), (T15,T7,87,35,39)
16	Scheduler 3	(T26,T13,87,25,25), (T27,T13,87,27,28)	(T9,T4,87,5,8), (T5,T2,87,10,18)	(T28,T14,87,30,30), (T29,T14,87,32,33), (T15,T7,87,35,39)

Table 1. (continued)

17	Scheduler 3	(T26,T13,87,25,25), (T27,T13,87,27,28)	(T9,T4,87,5,8), (T5,T2,87,10,18)	(T29,T14,87,32,33), (T15,T7,87,35,39)
18	Scheduler 2	(T26,T13,87,25,25), (T27,T13,87,27,28)	(T18,T9,87,5,5), (T19,T9,87,7,8), (T5,T2,87,10,18)	(T29,T14,87,32,33), (T15,T7,87,35,39)
19	Scheduler 1	(T27,T13,87,27,28)	(T18,T9,87,5,5), (T19,T9,87,7,8), (T5,T2,87,10,18)	(T29,T14,87,32,33), (T15,T7,87,35,39)
20	No more Transitions are active	(T27,T13,87,27,28)	(T18,T9,87,5,5), (T19,T9,87,7,8), (T5,T2,87,10,18)	(T15,T7,87,35,39)

8 Discussion

We have simulated the model several times. Each time we start a new simulation, we got different series of executed transitions, however, all these simulations led to the same final result (status 20). The value 87 has been found in the location 33 by thread number 29 (status 19). In addition, in this trial of simulation, only two stealing incidents happened (statuses 2 and 4). The results of simulation show consistency towards reaching the ending thread. No errors have been detected nor any kind of ambiguity appeared during the execution time. Although CPN-Tool is a single thread tool in the sense that the tool has to move from one transition to another, the concurrent execution of the model appeared clearly in the results. The concurrent execution happened in the statuses 2-19. It is obvious that in each of the statuses 2-19, we have more than one transition that can be executed concurrently.

Design simplicity and scalability are other significant achievements in this study since it is easy to expand the model by adding new cores. Only a slight change may be needed to the Enhanced Work Stealing code to include the new cores.

CPN-ML as a functional language uses lists as their main data structure. The language is supported by a huge number of list's built-in functions that process the list and return results in a short time. In addition, CPN-ML is free of side effects as in imperative languages. Eliminating side effects makes the behaviour of a program much understandable and predictable.

The internal structure of the stack's list and thread's list proves its robustness in processing the threads. CPN-ML, as a functional language derived from SML, uses linked list in creating lists. Using linked lists has several advantages, as in the following, comparing with other data structures such as arrays.

First, using fixed array suffers from the problem of overflow as in previous studies even so when circular arrays have been used to solve this problem. Managing the indices and sometimes copying data from one region to another to spare more space causes a lot of overheads. However, in functional languages, there are no such problems since the entire space is under the control of the language except when the total required space exceeds the size of the memory.

Second, memory management in functional languages is much easier than in other languages like C and C++. Modelling using CPN-ML releases the designer from any kind of memory management problems. The mechanism of garbage collector is responsible for returning areas of memory that are no longer in use. The lack of such mechanism in languages such as C and C++ adds more overhead since codes have to be included to reorganize the memory.

In our study, we used stacks as local memories comparing with using queues or dequeues as in other studies. We found that using a stack has several useful characteristics: First it is much easier to deal with one end instead of two (less code). Second, for reasons related to security and management, stacks are preferable than queues. Finally, as we used CPN-ML as a language of modelling, representing a stack as a list enables us to use all the lists' built-in functions.

The Enhanced Work Stealing algorithm plays the main role in balancing threads' load among the cores. The algorithm has a privilege comparing with other Work Stealing techniques which is the ability to make more than one core steal at the same time (one step). In case we add more cores, the Enhanced Work Stealing has to be expanded slightly to deal with the new expansion. It is likely to have more than one idle core and more than one non-empty core. The pattern matching technique allows the redistribution of loads (threads) among the cores to be performed in one step in a fair manner.

9 Conclusion

This paper presents a new model for solving one of the Divide and Conquer problems, i.e. Binary Search. The model has been designed to be adaptable to a multicore environment through concurrent multithreaded scheduling. A new algorithm in addition to one enhanced algorithm has been designed to support the work of the designed model. Binary Search Multithreaded Scheduling (BSMS) algorithm is designed to control the creation and movement of threads within each modelled core. BSMS function is responsible for the dividing part of the Divide and Conquer (D&C) general strategy. The Enhanced Work Stealing, EWSS, is designed to control threads distribution among the modelled cores. The EWSS function is responsible for the concurrent part in this model. Scalability, accuracy, simplicity and fair distribution of load are the main characteristics of the model. The simulation process of the model proved its correctness and stability. The results of simulation show effective concurrent execution of the model. In many cases, more than one modelled core can be executed at the same time.

In this study, we succeeded in solving several shortcomings happened in the previous studies such as limitation of the number of threads, overflow, side effects, language complexity, indices and memory management. Concurrent execution and fair distribution of threads among the cores were the main targets in this research. However, our policy in controlling threads movement does not take into consideration the richness of the cores when we want to redistribute the load.

References

1. Cormen, T., Leiserson, C., Rivest, R.: Introduction to Algorithms. MIT Press, Cambridge (2000)
2. Levitin, A.: Introduction to the Design and Analysis of Algorithms. Addison Wesley, Reading (2002)
3. Dasgupta, S., Papadimitriou, C., Vazirani, U.: Algorithms. McGraw-Hill, New York (2006)
4. Harris, S., Ross, J.: Beginning Algorithms. Wiley, Chichester (2006)
5. Kavi, K., Moshtaghi, A., Chen, D.: Modeling Multithreaded Applications Using Petri Nets. International Journal of Parallel Programming 30(5) (October 2002)
6. Jensen, K., Kristensen, L.: Coloured Petri Nets, Modelling and Validation of Concurrent Systems. Springer, Heidelberg (2009)
7. Jensen, K.: An Introduction to the Practical Use of Coloured Petri Nets, vol. 1492, pp. 237–292. Springer, Heidelberg (1998)
8. Kristensen, L., Christensen, S.: Implementing Coloured Petri Nets Using a Functional Programming Language. Higher-Order and Symbolic Computation 17(3), 207–243 (2004)
9. Mulyar, N., Aalst, W.: Patterns in Colored Petri Nets. Beta Working Paper Series (2005)
10. CPN-Tool Homepage, <http://cpntools.org/>
11. Blumofe, R., Leiserson, C.: Scheduling Multithreaded Computations by Work Stealing. Journal of the ACM 46(5), 720–748 (1999)
12. Arora, N., Blumofe, R., Plaxton, C.: Thread Scheduling for Multiprogrammed Multiprocessors. In: The Tenth Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA), Puerto Vallarta, Mexico, pp. 119–129 (1998)
13. Hendler, D., Lev, Y., Moir, M., Shavit, N.: A Dynamic-Sized Non-blocking Work Stealing Deque. Journal of Distributed Computing 18(3), 189–207 (2005)
14. Hendler, D., Shavit, N.: Non-blocking Steal-Half Work Queues. In: The Twenty-First Annual Symposium on Principles of Distributed Computing, Monterey, California, pp. 280–289 (2002)
15. Acar, U., Blecloch, A., Blumofe, R.: The Data Locality of Work Stealing. In: The Twelfth Annual ACM Symposium on Parallel Algorithms and Architectures, Bar Harbor, Maine, United States, pp. 1–12 (2000)
16. Chase, D., Lev, Y.: Dynamic circular work-stealing deque. In: The Seventeenth Annual ACM Symposium on Parallelism in Algorithms and Architectures, Las Vegas, Nevada, USA, pp. 21–28 (2005)
17. Peterson, J.: Petri Nets. ACM 9(3) (1977)
18. Murata, T.: Petri Nets: Properties, Analysis and Applications. IEEE 77(4), New York (1989)
19. Gansner, E., Reppy, J.: The Standard ML Basis Library. University of Cambridge, New York (2002)
20. Ullman, J.: Elements of ML Programming. Prentice-Hall, Englewood Cliffs (1998)

Appendix: CPN-ML Code of the Binary Search Multithreaded Concurrent Model

```

colset INT = int; colset BOOL = bool; colset STRING = string;
colset IntList = list INT; var LInt: IntList; colset ThreadId= STRING;
colset ThreadFather = STRING; colset Element = INT; colset Starting= INT;
colset Ending= INT;

colset Thread = product ThreadId * ThreadFather * Element * Starting * Ending;

```

```

var thr : Thread; colset List_of_Threads = list Thread;
var LstThrIn, LstThrOut, C1Out, C1In, C2Out, C2In, C3In, C3Out: List_of_Threads;
var LocIn, LocOut: INT; var ContIn, ContOut: BOOL;
fun Right_Child_Id (p1)="T"^Int.toString(valOf(Int.fromString
                                (String.extract (p1, 1, NONE))) * 2 + 1);
fun Left_Child_Id (p1)="T"^Int.toString(valOf(Int.fromString
                                (String.extract (p1, 1, NONE))) * 2);
fun leaf (S:Starting, NewR, NewC, Elem, V, lot) =
  let
    val NR = if (Elem = List.nth (V, S)) then S else ~1;
    val NC = if (Elem = List.nth (V, S)) then true else NewC;
  in (NR, NC, V, lot)
  end;
fun semi_leaf (S:Starting, NewR, NewC, Elem, V, lot) = let val NR = if (Elem = List.nth (V, S))
                                then S else if (Elem = List.nth (V, S+1)) then (S+1) else ~1;
val NC = if (Elem = List.nth (V, S)) then true else if (Elem = List.nth (V, S+1)) then true
                                else NewC; in (NR, NC, V, lot) end;

fun others (S:Starting, E:Ending, Id, NewR, NewC, Elem, V, lot) =
let
  val middle = (S+E) div 2;
  val NR = if (Elem = List.nth (V, middle)) then middle else ~1;
  val NC = if (Elem = List.nth (V, middle)) then true else NewC;
  val NewLot = if (NC=false) then [(Left_Child_Id (Id), Id, Elem, S, middle - 1)]^^
[(Right_Child_Id (Id), Id, Elem, middle + 1, E)]^^ lot else lot;
in (NR, NC, V, NewLot)
end;

fun BSMS(V: IntList, lot: List_of_Threads, Cont, Res)=let val h = hd lot;
val lot= List.drop(lot, 1);
val Id = #1 h; val Father = #2 h; val Elem=#3h; val S = #4 h; val E = #5 h; val NewR = 0;
val NewC = false;
val (NewR, NewC, V, lot)= if (S=E) then leaf (S, NewR, NewC, Elem, V, lot) else if (S+1=E) then
semi_leaf (S, NewR, NewC, Elem, V, lot) else others (S, E, Id, NewR, NewC, Elem, V, lot)
in (NewR, NewC, V, lot)
end;

fun Guardian(a,b,c) = if (length a = 0 orelse length b = 0 orelse length c = 0) andalso
(length a > 1 orelse length b > 1 orelse length c > 1) then true else false;

fun EWSS([], [], []) = ([], [], [])
|EWSS([], [], a::[]) = ([], [], a::[])
|EWSS([], [], a::(b::c)) = (a::[], b::[], c)
|EWSS([], a::[], []) = ([], a::[], [])
|EWSS([], a::[], b::[]) = ([], a::[], b::[])
|EWSS([], a::[], c::(d::e)) = (c::[], a::[], d::e)
|EWSS([], a::(b::c), []) = (a::[], c, b::[])
|EWSS([], a::(b::c), d::[]) = (a::[], b::c, d::[])
|EWSS([], a::(b::c), d::(e::f)) = (a::[], b::c, d::(e::f))
|EWSS(a::[], [], []) = (a::[], [], [])
|EWSS(a::[], [], c::[]) = (a::[], [], c::[])
|EWSS(a::[], [], c::(d::e)) = (a::[], c::[], d::e)
|EWSS(a::[], c::[], []) = (a::[], c::[], [])

```

$\text{IEWSS}(a::[],c::[],e::[])=(a::[],c::[],e::[])$
 $\text{IEWSS}(a::[],c::[],e::(f::g))=(a::[],c::[],e::(f::g))$
 $\text{IEWSS}(a::[],c::(d::e),[])=(a::[],d::e,c::[])$
 $\text{IEWSS}(a::[],c::(d::e),f::[])=(a::[],c::(d::e),f::[])$
 $\text{IEWSS}(a::[],c::(d::e),f::(g::h))=(a::[],c::(d::e),f::(g::h))$
 $\text{IEWSS}(a::(b::c),[],[])=(c,a::[],b::[])$
 $\text{IEWSS}(a::(b::c),[],d::[])=(b::c,a::[],d::[])$
 $\text{IEWSS}(a::(b::c),[],d::(e::f))=(a::(b::c),d::[],e::f)$
 $\text{IEWSS}(a::(b::c),d::[],[])=(b::c,d::[],a::[])$
 $\text{IEWSS}(a::(b::c),d::[],f::[])=(a::(b::c),d::[],f::[])$
 $\text{IEWSS}(a::(b::c),d::[],f::(g::h))=(a::(b::c),d::[],f::(g::h))$
 $\text{IEWSS}(a::(b::c),d::(e::f),[])=(b::c,d::(e::f),a::[])$
 $\text{IEWSS}(a::(b::c),d::(e::f),g::[])=(a::(b::c),d::(e::f),g::[])$
 $\text{IEWSS}(a::(b::c),d::(e::f),g::(h::i))=(a::(b::c),d::(e::f),g::(h::i));$

An Approach for Source Code Classification Using Software Metrics and Fuzzy Logic to Improve Code Quality with Refactoring Techniques

Pornchai Lerthathairat and Nakornthip Prompoon

Software Engineering Lab, Center of Excellence in Software Engineering,
Department of Computer Engineering,
Faculty of Engineering, Chulalongkorn University, Thailand
pornchai.lerthathairat@thomsonreuters.com,
Nakornthip.S@chula.ac.th

Abstract. The problem of developing software quality is developer's experience which has different style and does not care about coding with principle that causes error or even bad smell. To reduce the risk of causing bad smell, the developer should concern with a good design principle and coding well. In addition, knowing the qualification and the characteristic of code is also important to promptly support verifying, recovering bad smell and improving them to be a good code. This research presents an approach for source code classification using software metrics and fuzzy logic to improve code quality with refactoring techniques. Our approach composed of 3 main sections; Source code definition with metrics and evaluation to classify source code type, Source code classification with fuzzy logic and Source code improvement with refactoring. The result of our approach is able to classify source code in type correctly and improve bad smell, ambiguous code to be a clean code.

Keywords: Code Quality Improvement, Clean Code, Fuzzy Logic, Refactoring, Software Metrics.

1 Introduction

In order to get software quality by Phillip Crosby's principle "Conformance to requirement"[15], design and programming process is an important step about software development but the problems occur repeatedly during developing, for example, analysis requirements inaccurately, time pressure or even the developer's experience that coding in different style. These problems generate poor design and bad smell [2] that needs to correct them before delivering to the users. One technique of Bad smell resolution is refactoring which is a method of elimination bad smell.

In fact, the developer tries to design and do programming with good code or clean code. Clean code can help internal software working effectively and indicate a good design. By the characteristic of clean code that Robert C. Martin [16] explained; it should be simple, well-written, easy to enhance. When the developer measure and evaluate the quality, the result will show the different between clean code and bad

smell that the developer should concentrate on coding with clean code from the start rather than eliminate bad smell at the end.

This research presents an approach for source code classification using software metrics and fuzzy logic to improve code quality with refactoring techniques that divided into 3 main sections; first section is source code definition with metrics and evaluation, the method classifies source code with software metrics to categorize the code types. After classifying, we have found an ambiguity between clean code and bad smell that we define as ambiguous code. The second section is source code classification with fuzzy logic because the ambiguous code cannot classify or measure with metrics. We have to bring fuzzy logic to clarify the ambiguity by using knowledge's expert to set rule base, for example, bad smell gets rule base from analysis Mika Mäntylä's research [12]. The last section is source code improvement with refactoring. The method improves bad smell and ambiguous code with appropriated refactoring techniques until they are verified to be clean with clean code criteria. The scope of this research is only classifying source code and selects the improvement techniques suitably in order to be clean code with quality but does not measure the value of quality in each code type.

Section 2 briefly summarizes background knowledge relevant to this approach, section 3 is related works, and section 4 is the approach and the methodology. The last section provides a conclusion and future works.

2 Knowledge Background and Related Works

2.1 Design Principle and Source Code Classification

According to a good design and programming [1, 19, 24] the developer should understand that each object in software can interact and collaborate of each other. Object-oriented programming (OOP) is a programming paradigm base on the idea of using the flexible objects work together to enhance programming capability with quality. To accomplish the objectives, there are five important principles of programming [16]; Single Responsibility Principle (SRP), Open/Closed Principle (OCP), Liskov Substitution Principle (LSP), Dependency Inversion Principle (DIP) and Interface Segregation Principle (ISP). Therefore, to write code with the design principle will be possible to conduce a clean code but if it does not follow the principles. They would produce a bad smell instead. Our approach uses the principle for defining clean code and the rule base for defuzzication in a classification process.

The source code classification with software metrics defines the code type into 2 groups; bad smell and clean code. However after doing a research, we have found another type which is an ambiguity. It lines between clean code and bad smell. We introduce it as an ambiguous code. Its definition is in the following details.

2.1.1 Bad Smell

This kind of code does not work in the right qualification. [12, 25] The reason of causing bad smell for example, the developer's experience which is different styles, requirements changed frequently and the external facts which constrain the design and programming process more shorten. The developer does not have time to think about

coding with the design principle, finally the complexity of code and bad smell that cause software quality degradation. The developer must pay an attention to verify and find a solution to correct it. There are 22 bad smell types for an example, Long method has large size and takes many responsibilities that hard to understand and modify, Large Class is a class that tries to do many jobs in software. These classes have too many instance variables or methods.

Our approach uses the characteristic of bad smell type to define bad smell criteria and the rule base for defuzzication in the classification process then improves it to be clean code with refactoring techniques.

2.1.2 Clean Code

A kind of code that helps software working smoothly because of its characteristic follows the design principle [16], for example, readability, none of the duplication. In additional, clean code help decreasing risk when the requirement changed, guiding the developer to get a good design and programming to reduce the amount of corrections. There are more 80 clean code types for an example; Function should be small and do one thing, it is easy to understand and flexible to change, Meaningful Names is the name of a variable, function, or class, should answer all the purpose. It should tell the developer why it exists, what it does, and how it is used. If a name requires a comment, then the name does not reveal its intent.

So, our approach uses the characteristic of clean code to define clean code criteria and the rule base for defuzzication in the classification process.

2.1.3 Ambiguous Code

After classifying source code with metrics, we have found an unspecified code type. It cannot measure by metrics same as clean code and bad smell but the ambiguity between clean code and bad smell need to clarify and categorize it into the correct group because it is necessary to improve bad smell and some ambiguous code to be clean code

Therefore, classifying three groups of source code, our approach uses software metrics to present the qualification of each code and also ensure that classification method works precisely.

2.2 Software Metrics

Our research uses software metrics because it is a tool of software measurement with Mathematics and science [22, 13]. The approach uses McCabe, Kemerer and Halstead [11, 17, 18, 20] to set the specification of outlier to measure and to classify source code. But there is the limitation of using metrics is not able to classify the ambiguity between clean code and bad smell because of high complexity and does not support non-ambiguous then we have to find other techniques to clarify the ambiguity.

2.3 Fuzzy Logic

Fuzzy logic [7, 8, 26] is derived from fuzzy set theory to deal with reason and imitate the expert decision where binary set two-valued logic. Fuzzy logic ranges in degree (Set membership values) between 0 and 1. The approach uses Lotfi Zadeh concept

and techniques [5, 6, 7, 8, 9] to classify ambiguous code. When the method uses software metrics to measure and classify source code. There is some ambiguous code, which cannot classify because its qualification is very complex. We need to clarify and make a good decision under complicated conditions with If-Then form that conforms to human logic. The result from using fuzzy logic does not value only true or false but show the relation between true and false.

3 Our Approach

Our approach with aim to classify source code with software metrics and fuzzy logic to improve code quality with refactoring techniques is divided into 3 main sections as shown in Figure 1.

First section is source code definition with metric measurement and evaluation to define the code criteria for classification with software metrics. Second section is source code classification with fuzzy logic to classify three groups of code with rule base and clarify the ambiguity with fuzzy logic. The third section is source code improvement with refactoring to improve bad smell and ambiguous code with refactoring techniques until they are clean. Then we verify all the method for affirmative. The detail of each section is described in following detail:

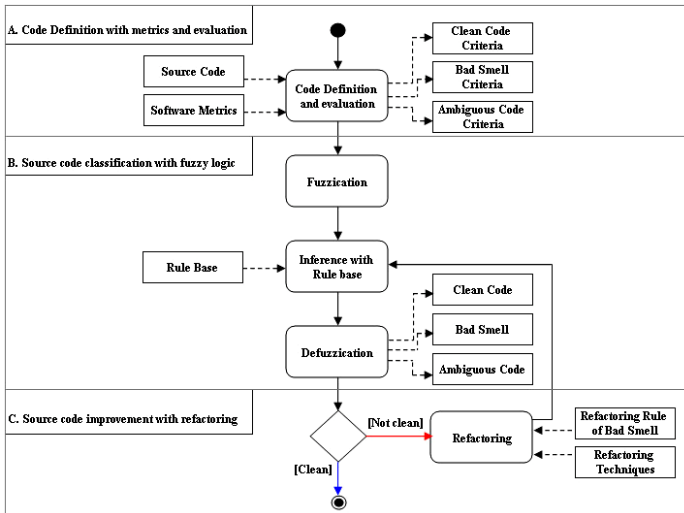


Fig. 1. An approach for source code classification using software metrics and fuzzy logic to improve code quality with refactoring techniques

3.1 Code Definition with Metric Measurement and Evaluation

In this section, to classify bad smell, clean code and ambiguous code. We have to prepare the classification rules with software metrics and the specification of outlier.

The method organizes two groups of source code sample; first set of source code for setting fuzzy rules and second set of source code for verification.

3.1.1 Bad Smell Selection Criteria

To select bad smell for our approach, we use Mäntylä's suggestion [12] which ranked bad smell group from level 0-5. Level 0 means bad smell detection is very complex that cannot use metrics to detect. This level needs only the expert to detect them. Level 5 means bad smell detection is easy to detect with few metrics. Our approach selects level 4 – 5 which focus only the Bloaters group. The Bloater is something that has grown large that it cannot be effectively handled.

Therefore, by the bad smell criteria; we detect it with an semi auto-system that does not count on any expert. So bad smell which qualified are the Long method, Large class and Long parameter list.

3.1.2 Clean Code Selection Criteria

To select clean code for our approach, we impose five conditions;

- Code is explicit and readable. The developer can understand the qualification of code, where code link to and the output.
- Code can be explained with software metrics that means code is written with OOP principle [16] and can be measurable [13, 22].
- Code does not develop from the engine. The developer should analyze and plan. They should not count on only the machine because when there is any problem happened. They should know the cause immediately.
- Code is flexible and portability that will accelerate the software cycle.
- Code is not duplicated because one object should respond for only one responsibility. If it is repeated, the developer should re-categorize to prevent duplication.

For our approach, we choose Function, Classes and Comment because function and class are important in programming. It has to be well-structure and comment is necessary involved to describe the responsibility purpose of function and class.

3.1.3 Source Code Measurement with Software Metrics

To select metrics for Source code measurement, we use basic software metrics [11, 17, 18, 20]. We set the specification of outlier to classify source code, as the following tables. (Software metrics Acronym is available at Appendix)

3.1.3.1 Metrics and the Specification of Outlier of Bad Smell. According to the bad smell criteria, we bring Mäntylä's suggestion [12] to set the specification of outlier is shown in the table 1. In the table, we use NLOC to measure Long Method. The specification of outlier is equal to or more than 60. If a source code is evaluated and NLOC more than or equal to 60 that means it is in a bad smell type.

Table 1. Metrics and the specification of outlier of Bad smell

Bad Id	Description	Metrics (m)	The specification of outlier (x)
1	Long Method	NLOC(1)	$x \geq 60$
		NILI(2)	$x \geq 200$
		CC(3)	$x \geq 30$
		ILCC(4)	$x \geq 60$
2	Long Parameter List	NOP(5)	$x > 7$
3	Large Class	LCOM(8)	$x > 0.8$
		LCOM-HS(9)	$x > 1.0$
		NFD(7)	$x > 20$
		NOM(6)	$x > 20$

3.1.3.2 Metrics and the Specification of Outlier of Clean Code. The metrics for clean code classification, we analyze from the idea proposal by Robert and software experts [16] to choose metrics accord with qualified clean code by criteria as the table 2. In the table, we measure Small Function by using NLOC and the specification of outlier that equal 20 or less. If source code is evaluated and NLOC is less than or equal to 20 that means it is in a clean code type.

Table 2. Metrics and the specification of outlier of Clean code

Clean Id	Description	Metrics (m)	The specification of outlier (x)	Clean Id	Description	Metrics (m)	The specification of outlier (x)		
1	Comment	PCC	$x \geq 20$	9	Class Organization (Cont.)	NOIM	$x < 20$		
2	Small Functions	NLOC(1)	$x \leq 22$	10	Encapsulation	NMI	$x < 1$		
		NILI(2)	$x \leq 50$			11	Classes Should Be Small	NPF	$x < 1$
		CC(3)	$x \leq 15$					NOP	$x \leq 5$
		ILCC(4)	$x \leq 40$					NOV	$x \leq 8$
RC	$1.5 < x < 3.5$	NOO	$x \leq 6$						
3	Do One Thing	ABL	$x < 1.5$	12	Maintaining Cohesion	NOM(6)	$x \leq 10$		
4	One Level of Abstraction per Functions	ABL	$x < 1.5$			NFD(7)	$x \leq 10$		
5	Switch Statements	ILND	$x \leq 4$			LCOM(8)	$x \leq 0.5$		
6	Function Arguments	NOP(5)	$x \leq 5$			LCOM-HS(9)	$x \leq 0.8$		
7	Have No Side Effects	NOI	$x < 1.4$	13	Organizing for Change	ACML	$x > 0$		
8	Structured Programming	ISS	$x = 1$			ECML	$x \leq 50$		
9	Class Organization	NOC	$x < 6$						
		DIT	$x < 6$						

3.1.3.3 Metrics and the Specification of Outlier of Ambiguous Code. The method uses software metrics to help measuring and classifying source code as the previous section. We analyzed the specification of outlier and found some codes, which cannot describe what type of code. We defined them as an ambiguous code and try to use some metrics from table 1 and 2 to classify but the result is still ambiguity. In the table 3, we measure Ambiguity in method by using NLOC and the specification of outlier which evaluated more than 20 but less than 60 that mean it is in an ambiguous code because it does not present clean value and bad smell value.

Table 3. Metrics and the specification of outlier of Ambiguous code

Amb Id	Description	Metrics (m)	The specification of outlier (x)	Amb Id	Description	Metrics (m)	The specification of outlier (x)
1	Ambiguity in Comment	PCC	$x < 20$	9	Ambiguity in Class Organization	NOIM	$x \geq 20$
2	Ambiguity in Method	NLOC(1)	$22 < x < 60$	10	Ambiguity in Encapsulation	NMI	$x > 1$
		NILI(2)	$50 < x < 200$			NPF	$x > 1$
		CC(3)	$15 < x < 30$	11	Ambiguity in Classes	NOP	$x > 5$
		ILCC(4)	$40 < x < 60$			NOV	$x > 8$
3	Ambiguity in One Thing	RC	$1.5 > x > 3.5$			NOO	$x > 6$
4	Ambiguity in Abstraction per Functions	ABL	$x \geq 1.5$			NOM(6)	$10 < x \leq 20$
5	Ambiguity in Switch Statements	ILND	$x > 4$			NFD(7)	$10 < x \leq 20$
6	Ambiguity in Arguments	NOP(5)	$5 < x \leq 7$	12	Ambiguity in Cohesion	LCOM(8)	$0.5 < x \leq 0.8$
7	Ambiguity in Side Effects	NOI	$x \geq 1.4$			LCOM-HS(9)	$0.8 < x \leq 1.0$
8	Ambiguity in Structured Programming	ISS	$x < 1$	13	Ambiguity for Change	ACML	$x < 0$
9	Ambiguity in Class Organization	NOC	$x \geq 6$			ECML	$x > 50$
		DIT	$x \geq 6$				

After using metrics with first set of source code, the results show in the table 1, 2 and 3. The classification result is shown in the table 4. There is an ambiguity in code and cannot specify by using metric from bad smell or clean code.

Table 4. Classification Evaluation (Pre-Test)

No.	Metrics(m)	Clean	Ambiguous	Bad Smell
1	NLOC	16.4	42.1	78.4
2	NILI	45.8	176.5	344.1
3	CC	14.1	27.8	44.2
4	ILCC	32.1	51.3	112
5	NOP	2.6	4.2	5.5
6	NOM	7.6	10.2	12.3
7	NFD	6.4	9.2	11.9
8	LCOM	0.4	0.9	0.9
9	LCOM-HS	0.6	0.8	1.1

From the table 4, to measure clean code, bad smell and ambiguous code, we use nine metrics, which is shared among three groups of source code. The measurement result is from initiation source code samples that point out an ambiguous code section cannot be judged or put into bad smell type because there is clean code qualification combines in ambiguous criteria.

This point indicates that metrics is not troubleshooting for non-ambiguous, it is necessary to use another measuring technique that can decide and extend the clarity by using expert’s knowledge.

3.2 Source Code Classification Method with Fuzzy Logic

This part is for applying the fuzzy logic to classify bad smell, clean code and ambiguous code, there are following 3 subsections;

3.2.1 Fuzzification of Input Variable

This section brings the specification of outlier from bad smell, clean code and ambiguous code to create the relation by Sigmoidal membership function [3] to implicate membership input set. We also use Triangular membership function and Trapezoidal membership function to present the qualification of clean code, bad smell and ambiguous code. Then we use Mamdani-type inference to implicate the input set into 2 types; the first type is divided into 2 ranges are called less and more, the second type is divided into 3 ranges are called low, moderate and high as shown in Figure 2:

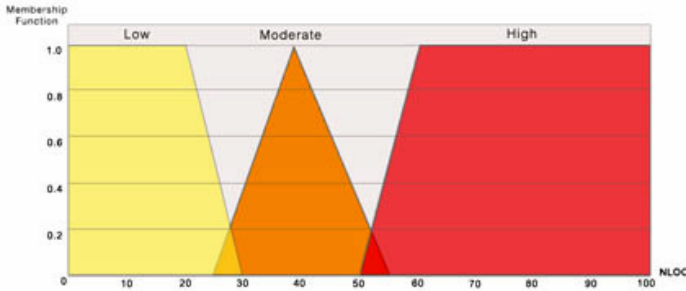


Fig. 2. Example of fuzzification input variable: NLOC

From the figure 2; NLOC is from all metrics and the specification of outlier and μ_x is Sigmoidal membership function, the fuzzification process converts the specification of outlier into input variable. This example presents input variable that shows the value as second type that divided into 3 ranges which are low moderate and high.

The fuzzification conduces to rule base which is a tactic in fuzzy model. The programming expert creates rule bases by forming the condition [3] which is completely true. We set the rule base by our programming experience more than 10 years.

3.2.2 Inference with Rule Base and Applying with Fuzzy Logic

The classification with fuzzy logic starts applying with rule base, as the following detail:

3.2.2.1 Bad Smell. This section analyzes and implicates bad smell with rule base; Bad smell type refers from rule condition as the table 5. In the table, Long method uses bad smell rule (NLOC is high) OR (NILI is high) OR (CC is high) OR (ILCC is high); according to the specification of outlier in table 1, the variable equal to 60 or more must be “High” and “OR” is an operation to present the maximum of variable along fuzzy logic that implicate as bad smell as show in the figure 2.

Table 5. Bad Smell Rules Formation

Rule No	Bad Smell Type	Rule condition
1	Long Method	(NLOC is high) OR (NILI is high) OR (CC is high) OR (ILCC is high)
2	Long Parameter List	(NOP is high)
3	Large Class	(LCOM is high) OR (LCOM-HS is high) OR (NFD is high) OR (NOM is high)
4		(NFD is high)
5		(NOM is high)

3.2.2.2 *Ambiguous Code.* When an ambiguous code is found, the process will analyze and also implicate with rule base. Ambiguous type refers from rule condition in the table 6. In the table, we use ambiguous rule condition to measure Small function; (NLOC is moderate) AND (NILI is moderate) AND (CC is moderate) AND (ILCC is moderate); according to the specification of outlier in table 3, the variable is between 20 and 60 is called “Moderate” that implicate as ambiguous code in the figure 2.

Table 6. Ambiguous Code Rules Formation

Rule No	Ambiguous Type	Rule condition	Rule No	Ambiguous Type	Rule condition
1	Comment	(PCC is less)	10	Encapsulation	(NPF is more)
2	Small Function	(NLOC is moderate) AND (NILI is moderate) AND (CC is moderate) AND (ILCC is moderate)	11	Classes Should Be Small	(LCOM is moderate) OR (LCOM-HS is moderate) OR (NFD is moderate) OR (NOM is moderate)
3	Do One Thing	(RC is low) OR (RC is high)	12		(NFD is moderate)
4	One Level of Abstraction per Functions	(ABL is more)	13		(NOM is moderate)
5	Switch Statements	(ILND is more)	14		(NOV is more)
6	Function Arguments	(NOP is moderate)	15		(NOO is more)
7	Have No Side Effects	(NOI is more)	16	Maintain Cohesion	(LCOM is moderate) OR (LCOM-HS is moderate)
8	Structured Programming	(ISS is less)	17	Organizing for Change	(ACML is less)
9	Class Organization	(NOC is more) AND (DIT is more) AND (NOIM is more) AND (NMI is more)	18		(ECML is more)

3.2.2.3 *Clean Code.* Clean code classification, the section is similar to previous section. Clean code will be analyzed and implicated with clean code rule base. Clean code Type is clean without any bad smell. If the process still finds some bad smells in source code, it should be clean up until it is clearly clean. The clean code type refers from rule condition as show in the table 7. In the table, we use clean code rule condition (PCC is Moderate) in comment; according to the specification of outlier in table 2, the variable equal to 20 or more must be “Moderate” that implicate as clean code as Figure 2.

Table 7. Clean Code Rules Formation

Rule No	Clean Code	Rule condition	Rule No	Clean Code	Rule condition
1	Comment	(PCC is more)	10	Encapsulation	(NPF is less)
2	Small Function	(NLOC is low) AND (NILI is low) AND (CC is low) AND (ILCC is low)	11	Classes Should Be Small	(LCOM is less) OR (LCOM-HS is less) OR (NFD is low) OR (NOM is low)
3	Do One Thing	(RC is moderate)	12		(NFD is low)
4	One Level of Abstraction per Functions	(ABL is less)	13		(NOM is low)
5	Switch Statements	(ILND is less)	14		(NOV is less)
6	Function Arguments	(NOP is low)	15		(NOO is less)
7	Have No Side Effects	(NOI is less)	16	Maintain Cohesion	(LCOM is less) OR (LCOM-HS is less)
8	Structured Programming	(ISS is more)	17	Organizing for Change	(ACML is more)
9	Class Organization	(NOC is less) AND (DIT is less) AND (NOIM is less) AND (NMI is less)	18		(ECML is less)

3.2.3 Defuzzication

After classifying a source code and apply with all rules, which are 5 rules of bad smell, 18 rules of ambiguous code and 18 rules of clean code. The next section is defuzzication by setting input set and applying with Center of gravity (COG) [3]. From the figure 3, it presents three possible situations of the qualification for each code as the following detail.

3.2.3.1 *Bad Smell.* From the Figure 3, the gravity value is represented on the right that means bad smell; first the method must improve bad smell with refactoring techniques until it is clean and then reduces ambiguous code.

3.2.3.2 *Ambiguous Code.* From the Figure 3, the gravity value is represented on the middle that means ambiguous code; it should be improved by refactoring techniques until it is verified to be clean. After the classification is completed, bad smell and ambiguous code have to be improved with refactoring to be clean code.

3.2.3.3 *Clean Code.* From Figure 3, after doing defuzzication, this method can approve a group of clean code is absolutely clean but permitted to have some ambiguous code. All bad smell and ambiguous code need to improve with refactoring techniques.

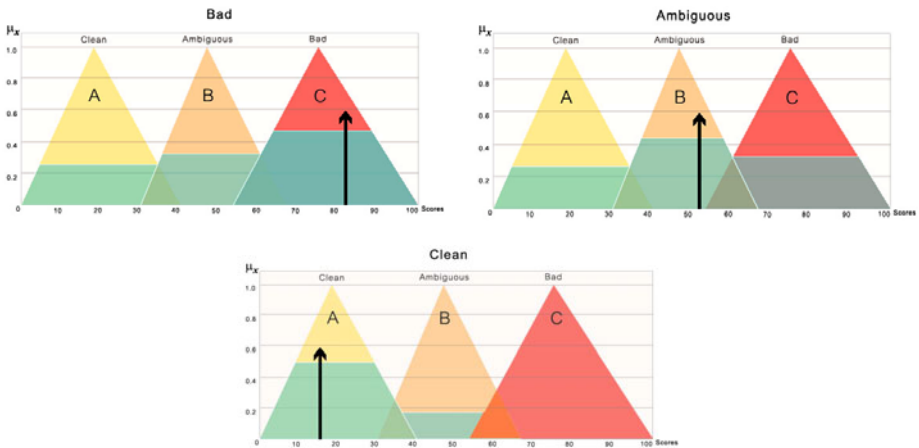


Fig. 3. Center of gravity value: Bad smell, Ambiguous Code and Clean Code

3.3 Source Code Improvement with Refactoring

This process emphasized on source code improvement. Every bad smell and ambiguous code has to improve with appropriated refactoring techniques by expert.

3.3.1 Improving Code by Clean Code Rules

To improve bad smell, we do it as the algorithm as the following;

Algorithm

```

Procedure ImproveSourceCodeToCleanCode(sourcecode)
begin
repeat
  codetype ← ClassifySourceCode(sourcecode);
  if (codetype is Bad Smell) then
    badsmell_type ← CheckBadsmellType(sourcecode);
    technique ← UseRefactoringTechniquefromTable(badsmell_type);
    sourcecode ← DoRefactoring(sourcecode, technique);
  else if (codetype is Ambiguous Code) then
    ambigous_type ← CheckAmbigousType(sourcecode);
    technique ← SuggestionFromProgrammingExpert(ambigous_type);
    sourcecode ← DoRefactoring(sourcecode, technique);
  end if
until (codetype is Clean Code or under acceptance criteria is true)
return {sourcecode}
end
    
```

By the algorithm, we classify source code. When we find bad smell, we use the refactoring techniques in the table 8 to improve it to be clean code. In additional, when we find source code as ambiguous code, we bring the suggestions from expert to select the appropriated refactoring techniques same as bad smell improvement until it is clean or the developer will decide to stop doing refactoring.

Table 8. Bad Smell Improvement with Refactoring Techniques

Refactoring Technique	Large Class	Long Method	Long Parameter List
Decompose Conditional		✓	
Duplicate observed data	✓		
Extract Class	✓		
Extract Interface	✓		
Extract Method		✓	
Extract Subclass	✓		
Introduce Parameter Object		✓	✓
Preserve Whole Object		✓	✓
Replace Method with Method Object		✓	
Replace Parameter with Method			✓
Replace Temp with Query		✓	

3.3.2 The Experiment Example of Our Approach

The experiment example of our approach presents source code improvement with appropriated refactoring techniques by expert which are represented in Section II Knowledge Background to bring bad smell on the right qualification as well as clean code.

From the table 9, to improve bad smell as the example; first step classify source code with bad smell rule base and we have found bad smell which are Long method and Long parameter list, so we select the suitable refactoring technique from table 8.

Second step presents the result after refactoring and classifying source code; we still have found bad smell which is Long parameter list that we have to improve it with refactoring again.

Third step presents that we can eliminate all bad smell but ambiguous code appear, they are Small Function, Do One Thing, Function Arguments and Have No Side Effects, so we have to select the appropriated techniques to manage them to be clean.

Fourth step presents that we can improve source code much better than the original source code but they are not clean as clean code criteria then we have to improve again by using knowledge's expert and different refactoring techniques which is Replace Parameter with Method.

The last step is measuring and evaluating source code. The result presents that almost source code is clean as clean code criteria. There is some of ambiguous code. We can accept because they do not impact to clean code qualification.

Therefore, refactoring can use for source code improvement. The method continuously cleans up bad smell and ambiguous code until they are verified to be a clean code without bad smell but the limitation of our approach is not able to count the refactoring cycle because some types of bad smell or ambiguous code are very complicated. Some cases, when improving bad smell might cause another bad smell. This is reason why cannot specify refactoring cycle certainly.

Table 9. The experiment example of source code improvement with refactoring techniques

Step	Classification Result	Metrics				Classification types	Refactoring techniques
1		NLOC	80.4	NOM	5.4	Long method	Extract Method
		NILI	311	NFD	5.5	Long parameter list	-
		CC	48	LCOM	0.3	-	-
		ILCC	121	LCOM-HS	0.5	-	-
		NOP	6.1			-	-
2		NLOC	44.7	NOM	5.4	Long parameter list	Introduce Parameter Object
		NILI	180	NFD	5.5	-	-
		CC	25.5	LCOM	0.3	-	-
		ILCC	48.7	LCOM-HS	0.5	-	-
		NOP	6.1			-	-
3		NLOC	42	NOM	5.4	Small Function	Replace Method with Method Object
		NILI	178	NFD	5.5	Do One Thing	-
		CC	22.2	LCOM	0.3	Function Arguments	-
		ILCC	40.1	LCOM-HS	0.5	Have No Side Effects	-
		NOP	5.4			-	-
4		NLOC	14.3	NOM	6.4	Function Arguments	Replace Parameter with Method
		NILI	48.2	NFD	6.2	Have No Side Effects	-
		CC	14.8	LCOM	0.3	-	-
		ILCC	34.7	LCOM-HS	0.5	-	-
		NOP	5.4			-	-
5		NLOC	14.0	NOM	6.6	Have No Side Effects	-
		NILI	47.8	NFD	6.2	-	-
		CC	14.2	LCOM	0.3	-	-
		ILCC	31.4	LCOM-HS	0.5	-	-
		NOP	2.4			-	-

3.4 Verification and Conclusion Process

This part is for analyzing the method and concluding a classification testing, there are 2 subsections as following;

Table 10. Comparison Source code: Pre-test and Post-test

Metrics (m)	The specification of outlier	Clean	Ambiguous		Bad Smell	
			Pre-test	Post-test	Pre-test	Post-test
NLOC	x <=22	16.4	42.1	19.2	78.4	19.4
NILI	x <=50	45.8	176.5	68.3	344.1	88.2
CC	x <=15	14.1	27.8	16.5	44.2	15.8
ILCC	x <=40	32.1	51.3	35.4	112	33.3
NOP	x <=5	2.6	4.2	3.1	5.5	2.9
NOM	x <=10	6.4	10.2	7.6	12.3	8.1
NFD	x <=10	6.6	9.2	8.7	11.9	7
LCOM	x <=0.5	0.4	0.8	0.5	0.9	0.5
LCOM-HS	x <=0.8	0.6	0.8	0.6	1.1	0.6

3.4.1 Verifying a Source Code

This section takes a source code for verification to affirm the correction of classification and source code improvement method.

After source code improvement process, every source code samples have to be compared between pre-test and post-test evaluation and present the code quality value which passes the classification method and refactoring. The value comparison is shown in the figure of source code: Pre-test and Post-test is shown in the table 10.

All 40 source code samples are classified with classification rules. The approach inspects and classifies clean code, bad smell, also specify an ambiguous code explicitly. Moreover bad smell and ambiguous code can be improved to be clean code but the quality cannot be better than developing code with good design and programming from the start.

4 Conclusion and Future Work

After, we classify source code with software metrics and fuzzy logic. We prove that we can classify three groups of source code; bad smell, ambiguous code and clean code. We can also improve bad smell and ambiguous code to be more quality by the programming expert. However, there is a limitation of improving source code with refactoring techniques that do not do it automatically by tool. The future work, hopefully to improve source code with refactoring automatically and use more source code samples to measure code quality with software metrics under software standards to build the code quality in depth.

References

1. Kay, A.C.: The Early History of Smalltalk, pp. 69–95 (1993)
2. Boehm, B.W., Basili, V.R.: Software Defect Reduction Top 10 List. *IEEE Computer* 34(1), 135–137 (2001)
3. Meesad, P.: Fuzzy Logic. Fuzzy systems and Neural Networks Lecture. Faculty of Information Technology, King Mongkut's University of Technology North, Bangkok
4. Stroggylos, K., Spinellis, D.: Refactoring-Does It Improve Software Quality? In: Stroggylos, K., Spinellis, D. (eds.) 5th International Workshop on Software Quality, vol. 10, IEEE Computer Society, Los Alamitos (2007)
5. Zadeh, L.A.: Fuzzy Sets. *Information and Control* 8, 338–353 (1965)
6. Zadeh, L.A.: Is There a Need for Fuzzy Logic? Fuzzy Information Processing Society, Annual Meeting of the North American 178(13), 2751–2779 (2008)
7. Zadeh, L.A.: Toward a Perception-Based Theory of Probabilistic Reasoning with imprecise probabilities. Special Issue on Imprecise Probabilities, *Journal of Statistical Planning and Inference* 105, 233–264 (2002)
8. Zadeh, L.A.: Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy Sets Systems* 90(2), 111–127 (1997)
9. Zadeh, L.A., Klir, G.J., Bo, Y.: Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems. In: *Advances in Fuzzy Systems - Applications and Theory*, vol. 6, World Scientific Pub Co. Inc, Singapore (1996)
10. Fowler, M.: Refactoring: Improving the Design of Existing Code. XP/Agile Universe (2002)
11. Halstead, M.H.: Elements of Software Science. In: *Operating and programming systems series*, Elsevier Science Inc., New York (1977)
12. Mäntylä, M., Vanhanen, J., Lassenius, C.: A Taxonomy and an Initial Empirical Study of Bad Smells in Code. In: *International Conference on Software Maintenance*, pp. 381–384 (2003)
13. Fenton, N.E.: *Software Metrics, A Rigorous Approach*. Chapman & Hall, London (1991)
14. Drucker, P.: *Innovation and Entrepreneurship*. Collins (1985)
15. Crosby, P.: *Quality is Free*. McGraw-Hill, New York (1979)
16. Martin, R.C.: *Clean Code. A Handbook of Agile Software Craftsmanship*. Prentice-Hall, Englewood Cliffs (2008)
17. Chidamber, S.R., Kemerer, C.F.: A Metrics Suite for Object Oriented Design. *IEEE Trans. Software Eng.* 20(6), 476–493 (1994)
18. Chidamber, S.R., Kemerer, C.F.: Towards a Metrics Suite for Object Oriented Design. In: *OOPSLA*, pp. 197–211 (1991)
19. Feldman, S.I., Kay, A.C.: A conversation with Alan Kay. *ACM Queue* 2(9), 20–30 (2004)
20. McCabe, T.J.: A Complexity Measure. *IEEE Trans. Software Eng.* SE-2(4), 308–320 (1976)
21. Ruhroth, T., Voigt, H., Wertheim, H.: Measure, Diagnose, Refactor: A Formal Quality Cycle for Software Models. In: *35th EUROMICRO-SEAA*, pp. 360–367 (2009)
22. DeMarco, T.: *Controlling Software Projects: Management, Measurement, and Estimates*. Prentice-Hall, Englewood Cliffs (1986)
23. Mens, T., Tourwé, T.: A Survey of Software Refactoring. *IEEE Trans. Software Eng.* 30(2), 126–139 (2004)
24. Alan Curtis Kay, http://en.wikiquote.org/wiki/Alan_Kay
25. Code smell, http://en.wikipedia.org/wiki/Code_smell
26. Fuzzy Logic, http://en.wikipedia.org/wiki/Fuzzy_logic
27. Software Quality, http://en.wikipedia.org/wiki/Software_quality

Appendix: Software Metrics Acronym

Acronym	Description	Acronym	Description
ABL	Abstraction Level	NLOC	Number Line of Code
ACML	Afferent Coupling at Method Level	NMI	Number of Methods Inherited
CC	Cyclomatic Complexity	NOC	Number of Children
DIT	Depth of Inheritance Tree	NOI	Number of Immutable
ECML	Efferent Coupling at Method Level	NOIM	Number of Interface with many Methods
ILCC	InLine Cyclomatic Complexity	NOM	Number of Methods
ILND	InLine Nesting Depth	NOO	Number of Overloads

Balanced Hierarchical Method of Collision Detection in Virtual Environment

Hamzah Asyrani Sulaiman¹ and Abdullah Bade²

¹ Universiti Teknikal Malaysia Melaka,
76100 Durian Tunggal, Melaka, Malaysia

² Universiti Malaysia Sabah,
88400 Kota Kinabalu, Sabah, Malaysia
asyrani@utem.edu.my, abade08@yahoo.com

Abstract. Determine the successful configuration between rigid bodies that come into contact is always a fundamental problem in computer visualization problem. Given the simulation that needs to maintain the speed of intersection, it is not always fast enough to compute the complete iteration between moving polyhedral. In this paper, we introduced a splitting algorithm that is able to successfully divide the hierarchical construction in virtual environment. By enhancing the capabilities of splitting rules of Bounding-Volume Hierarchies (BVH), the construction of BVH is improved and is able to provide fast collision detection method between two successive configurations.

Keywords: Collision Detection, Bounding-Volume Hierarchies, Algorithm.

1 Introduction

Virtual environments consist of various three-dimensional (3D) objects that are mimicking the real world properties such as trees, buildings, and many others. Most of these entities are considered as static object or rigid bodies where it cannot be deformed or alter when the simulation of virtual environment is running. In the simulation, however, few tasks need to be concurrently simulated with the virtual environment. Among these tasks are lighting, shadowing, texturing, culling, and collision detection. These tasks are considered as a realistic add-on where as in order to maintain the real-time virtual environment, these tasks need to be implemented in order to make the virtual environment becoming more realistic. If it is not, the simulated environment might become dull and cannot be realistic enough to attract the other people depending on what the application is targeted.

The realistic effect that has been put into virtual environment in order to make the simulation more interesting is collision detection. The term collision detection itself has been used by many other fields such as networking and medical where it involves component intersection checking. Collision detection is the critical component for simulated environment as it has been used to measure the realism between intersecting object in motion. Most researchers refer collision detection as an important tool for robotic, medical simulation, and computer games. Real-time

simulation always try to simulate the collision detection process as realistic as possible and thus the researchers have come out with numerous techniques in order to discuss between two or more intersected object. The collision detection consists of two parts which are discrete collision detection and continuous collision detection. Compared to continuous collision detection (CCD), discrete collision detection (DCD) is much faster in term of collision checking while CCD is more accurate. These two attributes cannot share the same advantages as increasing speed will eventually decreasing the accuracy of collision detection.

In this paper, we will introduce a splitting algorithm using Bounding-Volume Hierarchies (BVH) that works with DCD. The second section describes the previous work on collision detection. While the third section will explains the technique of BVH and Bounding-Volume (BV). The other two sections next after the third section which are fourth section and fifth section showed results and analysis from our work. Last section is the conclusion part of our implementation.

2 Previous Work

Significant amount of studies have revealed that collision detection between two objects can be divided into two phases which are broad phase and followed by narrow phase. [1] suggested that broad phase stands for the first phase of detecting object interference by checking which objects has collided. Next, narrow phase will be carried out to determine the exact collisions of both objects and which parts of this pairs has collided with detailed information.

Apparently, there are many types of algorithm to detect object interference in virtual environment that can be used in urban simulation. According to [1], these algorithms are; feature-based algorithms, simplex based algorithms, image-space based algorithms, volume based algorithms and spatial data structures such as BVH and space subdivision.

Feature-based algorithms intend to work directly with the primitives of the objects. Image space based algorithm is computed by image-based occlusion queries that usually implement on the graphics hardware (GPU). Volume based algorithms seem to work just like an image space based algorithm. However, for simplex based algorithm, it uses only the vertex of corresponding object information in order to construct a sequence of convex hulls [1]. One of the most popular simplex-based algorithms is GJK (Gilbert-Johnson-Keerthi) that becomes one of the most effective methods for determining intersection between two polyhedral [2] . In 1994, [3] presented exact collision detection to be used in large-scaled environments. Algorithm presented by [3, 4] used two types of Axis-Aligned Bounding-Boxes (AABB) which is fixed size boxes and dynamically-resized bounding boxes (dynamic boxes). They used Voronoi diagram to find a closest feature pairs. Here, they characterized the environments by the objects in motion and the complexity of the models. Regular virtual environment may require the simulation to give user satisfaction of being able to navigate through the virtual environment but that does not apply to the large-scaled environment such as urban simulation that has thousands of objects in virtual world.

Hence, performing accurate collision detection may consume long time just to check possible intersection area within objects in urban simulation. Thus leaving the only choice is to make sure that the collision detection technique work effectively.

There are another two types of collision detection method that widely been used which are BVH and space subdivision. Space subdivision intends to divide the spaces into small parts called cells but it not widely used for accurate collision detection. Example of space subdivision research can be found here [5, 6]. Bounding volume hierarchies provides more efficient technique using bounding-volumes that provides smaller and tighter hierarchies comparing to space subdivision [7-11].

3 Bounding-Volume Hierarchies

Bounding-Volume Hierarchies (BVH) is a hierarchical representation of 3D object in virtual environments to reduce the computational cost of in various applications such as culling system and collision detection. BVH are simply a tree structure that represents geometric models with specific bounding volumes. It works like a tree that has a root (upper division), a group of leafs (middle division) and a leaf (last division). Each node has it bounding-volumes that cover the children node. The main idea of BVH is to increase the level of the tree where it can create secondary node consists of left and right node. Each node stores a BV as a leaf node. Example of bounding-volume hierarchy is shown in Figure 1.

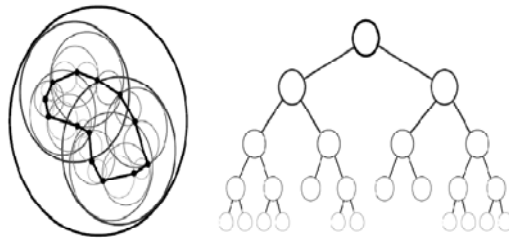


Fig. 1. The left hand side image shows a BVH with Sphere BV while on the right hand side image, shows unbalanced hierarchical form using binary type hierarchy

BVH allows the intersection occurs without searching for non-colliding pairs from the hierarchy tree. For example, given two objects with their BVH, when root of the hierarchies do not intersect, the calculation will not be performed for both objects. However, when the root of both hierarchies intersects, it checks for intersection between one root of the hierarchies' tree and the other children of objects hierarchies' tree. In this case, it recursively checks again whether there is intersection between both objects at middle level until it found the correct intersection.

4 Bounding Volume

Bounding-Volume (BV) is an important part of BVH construction. Numerous BV have been developed in the past in order to minimize the computational cost of performing collision detection. Instead of using primitive-primitive checking between intersected 3D objects, BV helps to speed up the process by enclosing bunch of triangles into single BV before proceed with collision checking. This is to reduce the possibility of eliminating set of triangles that does not intersect.

At the present time, there are several famous BVs such as spheres [12], Axis Aligned Bounding Box (AABB) [13-15], Oriented Bounding Box (OBB) [7, 15, 16], Discrete Oriented Polytope (k-DOP) [17], Oriented Convex Polyhedra [18], and hybrid combination BV [1]. Most large scale 3D simulations used bounding box because of the simplicity, require less storage, fast response of collision, and easy to implement [19]. Figure 2 illustrates most commonly used bounding volume.

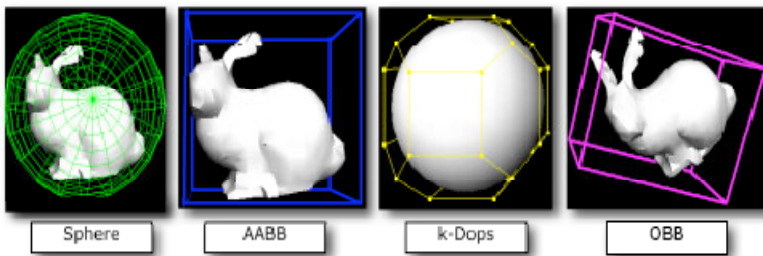


Fig. 2. Examples of common BVs as described in [18]

5 Our Algorithm

Our algorithm works by first enveloping the object with a simple BV called AABB. In order to create a hierarchical representation for the AABB, the AABB needs to be divided into two until it reaches certain level (defined by user or the program) or any stopping criteria. This division process is also called splitting process where it used selected splitting rules. The most common solution for splitting rules is to use Spatial Median technique. In this paper, the improvement has been made in order to enhance the capability of constructing fine and balanced hierarchical 3D object for collision detection. It also can be used to improve the accuracy of collision detection algorithm.

Spatial Median technique works by splitting the AABB using their maximum and minimum points. For example, if the maximum point for the corresponding object is in coordinate A (10, 0, 0) in 3D space, and the minimum point is in coordinate B (-10, 0, 0), then the splitting process will occurs at the coordinate $X = 0$. The splitting rules need to find all three axes and then perform a splitting process. The AABB will cut into almost half of its original size and the area is divided into two areas. Each area contains new points and need to recalculate their AABB and bound only the points according to its area.

Meanwhile for our method of splitting process is called Spatial Object Median splitting rules (SOMS). By using the midpoint of each triangle for the object, the object is then split into half using “a hidden AABB” that covered the midpoint. It is the same like having an AABB for all triangle midpoints. It is just the algorithm only needs to find the longest axis for separating plane of these midpoints AABB. In 3D world, object consists combination of triangles that can be calculated their midpoint. However, there are almost hundreds of midpoint type that can be calculated and our method can provides an efficient solution for any midpoint type that been used. The splitting process continues until it reaches certain level or stopping criteria that will stop the BVH construction. Figure 3 illustrates the new separating plane based on midpoint of triangles.

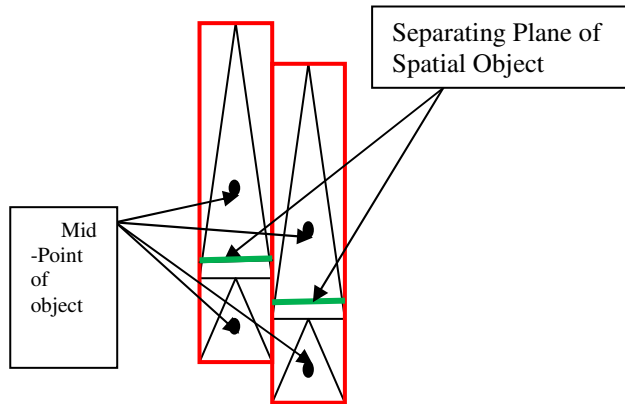


Fig. 3. An example on how the SOMS rule split the object triangle into two parts using the spatial midpoint between midpoints of two triangles which will become their separating plane

From Figure 3, it shows that new separating axis plane located between those two triangles midpoints. Thus, it could properly assign left and right nodes and store the corresponding triangle until it could have one BV one triangle. However for SOMS, it creates temporary spatial median of object median thus creating new median point.

Balance BVH tree is more efficient and fast when performing collision detection compared to unbalance BVH tree. Instead of faster construction using SOMS technique compared to Spatial Median, balance BVH tree helps reducing the potential of performing primitive-primitive testing on the earlier phase of detection. For example, given a node A with 40 triangles (using Spatial Median splitting rules and stop at level 7) and a node B with 20 triangles (using SOMS rules and stop at level 8). For each triangle of node A, it checks 40 times with the other object triangle. If the other object has 40 triangles too, it means that 40×40 tests must be done. However if we increase the level of BVH tree make it more balance just like node B with 20 triangles. It only needs to check 20×20 times for collision given each object while the construction time is similar. Although it is one level high compared to Spatial Median splitting rules technique, SOMS needs to perform only one test to move into the next BVH tree level for collision checking. Figure 4 depicts a short pseudo code for implementing this technique.

1. Start Create BV for the object
2. Calculate all midpoints of the object
 - a. Create Midpoints BV
 - b. Create BV space for all midpoints
 - c. Find Minimum and Maximum points for the midpoint space
3. Splitting Process
 - a. Determine the longest axis for separating plane
 - b. Split the BV (Midpoints BV) according to their spatial object median (SOMS)
4. Create Left and Right BV for the object using midpoint separating plane.
5. Repeat Procedure until Stopping Criteria is met.

Fig. 4. Algorithm to split the object using spatial object median (SOMS) and create BV according the midpoint separating plane

5 Implementation and Result Discussion

In this test, we developed an urban simulation representing the complex environments. An urban simulation is developed using 3D editor consists of 5672 triangles. The complex environment is built as a one large object consists of multiple buildings. The second object of Jet Plane is used for collision detection test using traversal algorithm. The number of triangles is explained in chapter three.

5.1 Tree Construction Time Test for Fixed Balanced Level of Spatial Median

Figure 5 shows that SOMS technique is faster than Spatial Median technique in term of construction time. SOMS is able to construct balanced BVH at fixed level 7 while Spatial Median become unbalanced when it reaches level 7. By analysing the Figure 5, we should notice that the BVH tree has been constructed multiple times (1000 times) and the average of these values is calculated.

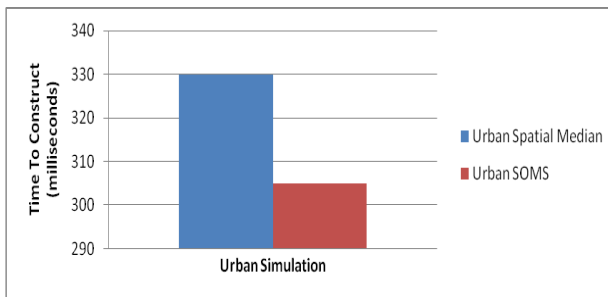


Fig. 5. Average times to construct 1000 times BVH for 3DS Urban Simulation using Spatial Median and SOMS method. The Level of BVH is fixed at level 6 (Spatial Median Balanced BVH).

Hence, if the object consists hundreds of thousands polygon, we can concluded that the construction time for real application is dropped when using SOMS technique compare to Spatial Median technique. Apart from that, the Figure 5 explained the number of leaf nodes for both techniques are not in the same BV size.

5.2 Tree Construction Time Test For Fixed Balanced Level of SOMS

In this experiment of Figure 6, Spatial Median technique generated less nodes compared to SOMS technique. SOMS BVH tree needs to produce balanced BVH level tree thus level 7 BVH tree is supposed to generate $2^7 = 128$ nodes. Each node is supposed to fill with triangles. However for Spatial Median technique, BVH that is generated has fewer nodes than the actual total nodes (less than 128 nodes). Hence, this proof that even though the results showed that SOMS much faster than Spatial Median, SOMS still be able to construct faster BVH tree with 128 nodes compared to Spatial Median technique.

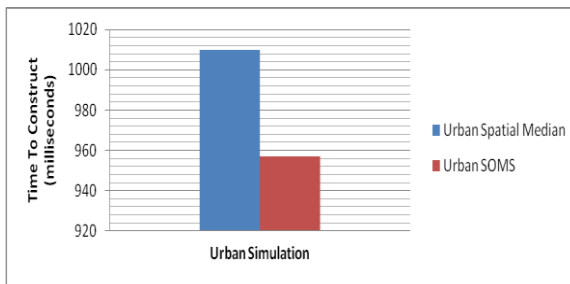


Fig. 6. Average times to construct 1000 times BVH for 3DS Urban Simulation using Spatial Median and SOMS method. The Level of BVH is fixed at level 7 (SOMS Balanced BVH)

In previous statement, we can see how number of triangles for certain objects is really important for SOMS technique to press its advantage. From the previous figure below calculation of Total Time in Minute for BVH tree construction, SOMS technique proved that for 1000 times construction test, it consumed 15 minutes and 57 seconds. Meanwhile for Spatial Median technique, the testing was running for 16 minutes and 50 seconds. The formula that we introduced here is to calculate the total time to construct 1000 times of BVH tree. It is our measurement to find BVH tree construction total time. From the testing that we run, we concluded that for any object that has hundreds of thousands polygon, SOMS be able to perform better than Spatial Median technique.

$$\begin{aligned}
 \text{Total Time in Minute (SOMS)} &= \frac{957 \times 1000 \text{ times}}{1000 \text{ milliseconds} \times 60} = \\
 &= 15 \text{ minutes and } 57 \text{ seconds} \\
 \text{Total Time in Minute (Spatial Median Splitting)} &= \\
 &= \frac{1010 \times 1000 \text{ times}}{1000 \text{ milliseconds} \times 60} = 16 \text{ minutes and } 50 \text{ seconds}
 \end{aligned}$$

From the calculation, it showed that SOMS technique produce a better BVH tree by achieving relatively less time compared to Spatial Median technique. As most of

the complex environments consisting hundreds of thousands polygon, the number of time taken to construct BVH tree is going to increase. Hence, the outcome of this testing is to measure how efficient BVH tree construction if the complex environments consist a lot of triangles.

6 Conclusion and Future Work

This technique intends to help reducing time to construct BVH while creating more balanced level or the tree reducing the use of heuristic determination. Since previous researchers used some heuristic to continue to split their tree, SOMS intends to reduce time to use heuristic for the node determination. This could benefit the real time construction when the real application of SOMS can be applied in deformable models instead of rigid bodies in future work

References

1. Kockara, S.H., Iqbal, I., Bayrak, K., Rowe, C.R.: Collision Detection - A Survey. In: Presented at the IEEE International Conference on Systems, Man and Cybernetics, ISIC 2007 (2007)
2. Bergen, G.V.D.: A fast and robust GJK implementation for collision detection of convex objects. *J. Graph. Tools* 4, 7–25 (1999)
3. Cohen, J.D., Lin, M.C., Manocha, D., Ponamgi, M.K.: I-COLLIDE: an interactive and exact collision detection system for large-scale environments. In: Presented at the Proceedings of the 1995 Symposium on Interactive 3D Graphics, Monterey, California, United States (1995)
4. Cohen, J.D., Lin, M.C., Manocha, D., Ponamgi, M.K.: Interactive and Exact Collision Detection for Large-Scale Environments. Technical Report TR94-005 (1994)
5. Subramanian, K.R., Fussell, D.S.: Applying space subdivision techniques to volume rendering. In: Presented at the Proceedings of the 1st conference on Visualization 1990, San Francisco, California (1990)
6. Bittner, J., Wonka, P., Wimmer, M.: Visibility preprocessing for urban scenes using line space subdivision. In: Pacific Graphics 2001 (Ninth Pacific Conference on Computer Graphics and Applications), pp. 276–284 (2001)
7. Chang, J.-W., Wang, W., Kim, M.-S.: Efficient collision detection using a dual OBB-sphere bounding volume hierarchy. *Computer-Aided Design* 42, 50–57 (2010)
8. Larsson, T.: Adaptive Bounding-Volume Hierarchies for Efficient Collision Queries, PhD, Mälardalen University, Sweden (2009)
9. Tuft, D.O.: A System For Collision Detection Between Deformable Models Built On Axis Aligned Bounding Boxes And Gpu Based Culling, Master of Science, Department of Computer Science, Brigham Young University, Brigham (2007)
10. Nguyen, A.: Implicit Bounding Volumes And Bounding Volume Hierarchies. Doctor of Philosophy, Stanford University (2006)
11. Sanna, A., Milani, M.: CDFast: an Algorithm Combining Different Bounding Volume Strategies for Real Time Collision Detection, *SCI*, pp. 144–149 (2004)
12. Liu, L., Wang, Z.-q., Xia, S.-h.: A Volumetric Bounding Volume Hierarchy for Collision Detection. In: 10th IEEE International Conference on Computer-Aided Design and Computer Graphics, pp. 485–488 (2007)

13. Zhang, X., Kim, Y.J.: Interactive Collision Detection for Deformable Models Using Streaming AABBs. *IEEE Transactions on Visualization and Computer Graphics* 13, 318–329 (2007)
14. Hubbard, R., Lin, M., Weller, R., Klien, J., Zachmann, G.: A Model for the Expected Running Time of Collision Detection using AABB Trees. In: *Eurographics Symposium on Virtual Environments (EGVE)*, Lisbon, Portugal (2006)
15. Tu, C., Yu, L.: Research on Collision Detection Algorithm Based on AABB-OBB Bounding Volume. In: *First International Workshop on Education Technology and Computer Science, 2009 ETCS*, pp. 331–333 (2009)
16. Gottschalk, S., Lin, M.C., Manocha, D.: OBBTree: a hierarchical structure for rapid interference detection. In: *Presented at the Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques* (1996)
17. Klosowski, J.T., Held, M., Mitchell, J.S.B., Sowizral, H., Zikan, K.: Efficient Collision Detection Using Bounding Volume Hierarchies of k-DOPs. *IEEE Transactions on Visualization and Computer Graphics* 4, 21–36 (1998)
18. Bade, A., Suaib. N., M. Zin. A., T. Sembok T. M.: Oriented convex polyhedra for collision detection in 3D computer animation, Presented at the Proceedings of the 4th International Conference on Computer graphics and Interactive Techniques in Australasia and Southeast Asia, Kuala Lumpur, Malaysia, 2006.
19. Lin, M.C., Manocha, D.: Collision and Proximity Queries. In: *Handbook of Discrete and Computational Geometry*, 2nd edn., vol. 35, pp. 787–807. CRC Press LLC, Boca Raton, FL (2004)

An Aspect Oriented Component Based Model Driven Development

Rachit Mohan Garg and Deepak Dahiya

Department of Computer Science and Engineering,
Jaypee University of Information Technology
Waknaghat, Solan, H.P, India

rachit.mohan.garg@gmail.com, deepak.dahiya@juit.ac.in

Abstract. This paper focuses on incorporating the concepts of aspects and software reuse in model driven architecture work for developing the highly reliable, adaptable software products in a timely fashion. This is achieved by partitioning the whole system into different independent components and aspects so as to facilitate component reuse along with the ease of modeling the components separately and emphasizing on the concerns that the widely used OOP paradigm has failed to address. Also it is supported to lay more emphasis on designing the software as compared to the traditional way of laying more emphasis on the coding than on modeling. Identification of reusable components is carried out using the hybrid methodology and aspects are identified by the domain experts. Along with the components, the PIM and aspects developed are stored in separate repositories so as to be used in development of other software of similar requirements and basic structure.

Keywords: MDA, System design, Platform Independent Model, Platform Specific Model, Component Based Software Development, Aspect Oriented Development.

1 Introduction

To survive the cut throat world of competition organizations have started spending most of their time in coding thereby resulting in a software that becomes less reliable, less adaptable since the actual and the hidden needs of the customer are not addressed completely. Moreover organizations are trying to develop the software in a cost effective way so they are using more of the available reusable components. The work presented in this paper describes a design methodology which will help in creating highly reliable, adaptable software products in a timely fashion.

A brief overview of the proposed methodology is as follows. It uses the concept of model driven architecture (MDA) [1], [2], [3], components [4], [5], [6], aspects [23], [24], [25]. Firstly the whole software is investigated or analyzed so as to reveal all the details. This includes the hidden or the inferred details along with the concerns corresponding to the software. Then the PIM and the aspects modeling are done separately using UML models. The aspects are modeled as separate entity so as to avoid tangling with the business logic. After the modeling, the PIM generated is

stored in the repository for reuse in other projects. Then the reusable components are identified from this optimized PIM using a hybrid identification approach. After the identification of the components, a repository of the components is maintained for future use. The aspects that are modeled separately are also stored in an aspect repository for future usage. The code for the aspects and software is generated in code generation and artifact generation phase respectively. They are weaved together to provide a complete application. Fig 1 represents the whole methodology with the help of a block diagram.

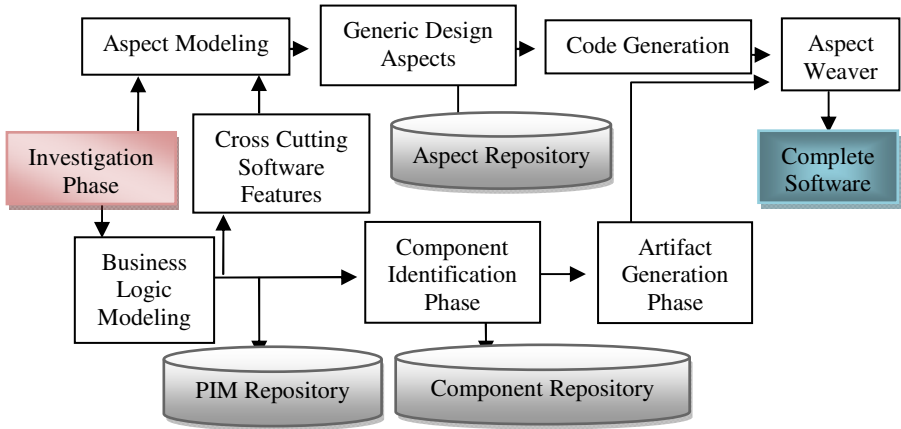


Fig. 1. Brief overview of the methodology

In this paper section 2 provides the literature review corresponding to integration of aspect oriented development, component based development and MDA. Section 3 describes the research methodology being proposed in this paper. Section 4 presents the results and significance of the methodology. The paper is concluded in section 5 followed by references.

2 Related Work

This section presents an overview of the related study in the field of aspects & components with MDA. First and the foremost a comparison and the problems associated with the different component identification techniques are provided. Then a study of the combination of components with MDA is provided followed by an overview of the UML profiles. Then a comparison of the different strategies providing combination of aspects with MDA is given.

2.1 Component Identification Techniques

Identification of reusable components in software is one the most important task of the component based software development. Many approaches for component identification are proposed but in the end it's the work of the experts to separate out

the components manually as these approaches only provide the knowledge of the component without actually separating them out.

Slicing Technique [14]. The slicing technique for identification of components is a family of techniques for isolating parts of a program according to the specified slicing criterion. It makes use of the control flow graphs (CFG) and program dependence graphs (PDG). By using both of these graphs the program is sliced into subprograms basically performing a specific function related to the overall software. Slicing techniques have been generally applied to the full-fledged software and not during the modeling phase as in the case of [14]. So an implementation of slicing technique or a variant of this methodology can be used during the modeling phase so as to figure out the desired components.

Fuzzy Clustering Method [17]. This method is basically used when there exist ambiguous boundaries among the clusters and objects belong to many clusters with a certain degree of membership. This allows the experts to discover new, undiscovered relations between the object and the corresponding clusters. FCM, one of the popular fuzzy clustering algorithm find a partition/clusters for a set of data points $x_j \in \mathbb{R}^d$, $j=1$ to N while minimizing the cost function. FCM suffers from the problem of not able to identify the initial partitions.

Clustering Techniques [18][19][20]. The proposed technique in [18] makes use of the concept that class relationships can be weighed according to their types. To identify business object component, the concept of resemblance degree between these objects is used. Resemblance degree depends on the relationship among the objects. The strength of edge considers both cohesion and coupling between business objects. Problem lies in the applicability of this technique to medium/big systems since it relies on the number of possible pairs of business objects.

The conversion of requirements into corresponding components is presented in [19]. This method makes use of the hierarchical clustering algorithm and defines an input matrix regarding the requirements which are later fed as inputs into the clustering technique.

2.2 Component Based Model Driven Development

In [21], a general discussion of applicability of CBSD in MDA is given but any approach to that is not depicted or showed. It just provides a possibility of the collaboration.

Not much research has been done on collaboration of CBSD with MDA. Most of the literature studied only provides the background of CBSD and MDA and whether they can be combined or not for effective software development.

2.3 UML Profiles [7][9]

In MDA, UML profile plays a very important role in expressing the PIM models, the PSM models, and the transformation rules. This profile can also be used as a semantic profile which enables model to express specific information. It can also be used for

tagging purpose so as to supply more information during model transformation and code generation. Figure 2 depicts inter-relation of UML and MDA.

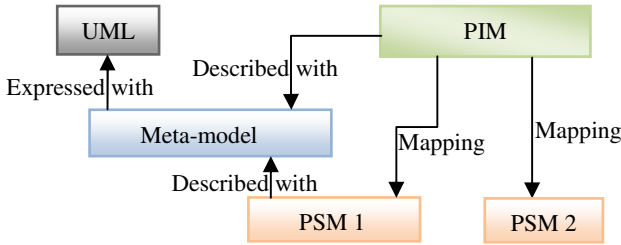


Fig. 2. Interrelation of UML and MDA

A UML profile is a combination of stereotype, tagged value and constraints. It uses stereotypes to assign special meaning to designated model elements i.e. how an existing meta-class may be extended. Whenever a stereotype is applied to a model element, this is shown by either an icon or the stereotype name between guillemets i.e. << >>. Figure 3 shows an example of stereotype.

When a stereotype is applied to a model element, the values of the properties may be referred to as tagged values. The profile constraints are used to specify the domain restrictions.

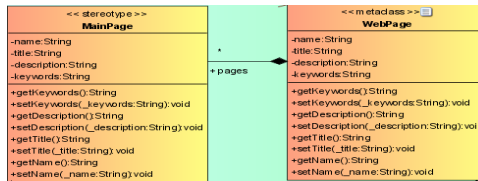


Fig. 3. UML profile

2.4 Integration of Aspects with Model Driven Development

The separation of concerns from the core business logic is a standard practice that helps in the development of better software that is free from the problem of scattering and tangling.

Concerns. The term *concerns* [9] doesn't reflect the issues related to the program, they represent the priorities and the requirements related to the software product. Concerns are basically divided into two broad categories which are further subdivided into other categories. Figure 4 gives a pictorial representation of the concerns that are discussed above.

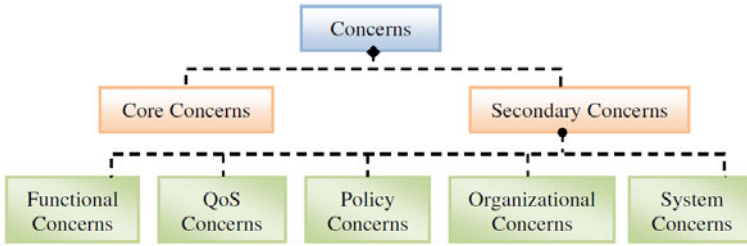


Fig. 4. Various types of Concerns

Jacobson [29][30] describes how the design aspects can be developed using use cases. It uses the concept of composition techniques that maps the work directly into the program level aspects. The proposed work does not provide details about models transformations, composition rules, structural relations, etc.

Reina [28] proposed an application of meta-models and UML profiles to separate out the concerns. The main problem addressed in the proposed methodology was that different meta-models were required for each of the concerns.

Mellor [31] proposed the integration of aspect oriented modeling with model driven architecture. They tried to design a framework to bring models and aspect together.

Kulkarni et al. [31] uses the concept of MDA to separate out the concerns from the business logic by using abstract templates.

Devon Simmonds [26][27] describes two approaches for integration of the aspects with the model driven development. The first approach Weave-then-Generate creates an integrated design model and from that model the code artifacts are generated. The second approach Generate-then-Weave creates a complete software application by the aspect code and the business logic code generated separately from the aspect model and primary model respectively.

3 Proposed Methodology

Now-a-days most of the organizations are using the UML diagrams only to fulfill the designing phase requirements and understand the flow of data. This was due to the fact that these models were only depicting the flow of data and control and had no relation with the coding.

The underlying proposed methodology incorporates the concept of component based development in model driven development. It also provides a repository based architecture in which the modeled PIM is stored so that it can be used in future project with the similar type of requirements and specifications along with the basic structure. Component Identification uses the process of *Forward Identification* [10] in which designers make use of the requirement models to identify business components (BC) and implement these BCs as Software Components (SC), then use these SCs to construct objective software systems. The generalized version of component identification process is depicted in figure 5.

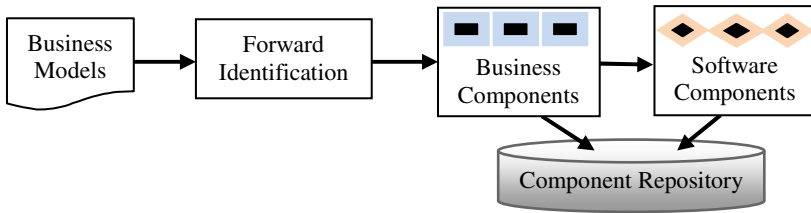


Fig. 5. Component Identification Process

Components are identified using a hybrid approach that uses the concept of both clustering and slicing. In a general term it is slicing followed by clustering so as to get the appropriate clusters that can be reused in other software products.

As opposed to the traditional approaches of software development this methodology along with the facility of the reuse of the developed PIM and the identified components, also provide support to the developers in coding by generating artifacts of the software for different platforms from the PIM created during design phase.

The proposed methodology is divided into different phases that perform specific tasks related to the design and development of the software.

3.1 Investigation Phase

This is the first and one of the most important phases of the methodology. As the name suggests this phase deals with the analysis of software under consideration. In this phase the software is analyzed so as to bring out each of its details. Many approaches are used to bring out these hidden features like questionnaire, meetings, holding interviews, holding focus groups etc. The compilation of the results of these different approaches helps in revealing all the features intended by the customer in software. Use case diagrams are one of the important tools used for showing the result of the proper analysis of the software's functionality in a graphical manner. It brings out many important details such as the intended users, their functionalities, interdependence of the functionalities of different users etc. Figure 6 represents the investigation process in a graphical way.

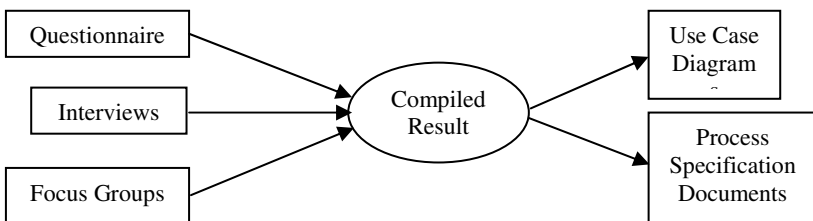


Fig. 6. Investigation Phase

3.2 Modeling Phase

This is the second phase of the proposed methodology that deals with the modeling of PIM. From the information collected in the investigation phase, the information regarding the aspects is separated from the core business requirements. Both the aspects and the core functionality are modeled in parallel. In this phase the corresponding UML diagrams viz. class diagram, state machine diagram, sequence diagram etc. for the software are modeled. The modeling is done in such a way that the diagrams don't include any information related to a specific platform or a language. The class diagram depicts different classes, interfaces, enumerations related to the software. Classes compose of the attributes and the related functionality for that particular class. Sequence diagram, State machine diagrams are used to depict the flow of control in a scenario or for a particular system or for a whole system as a whole. After the modeling of PIM information regarding the cross cutting software features revealed is provided to the aspect modeling so as to generate effective aspects. The modeled aspects are stored in an aspect repository so that the model of the aspects can be used in future. The PIM is stored in the PIM repository to facilitate the primary concept of MDA which is 'One Model Different Implementations'. The whole process of modeling phase is illustrated graphically in figure 7.

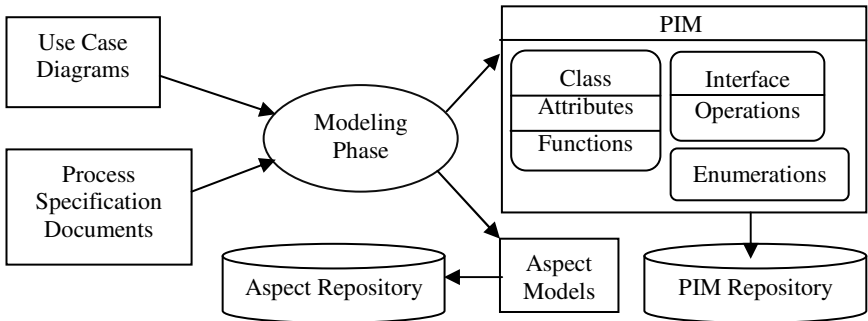


Fig. 7. Modeling Phase

3.3 Component Identification Phase

This is the third phase of the methodology that deals with the identification of the reusable components in the software. The proposed identification technique makes use of the advantages of slicing and clustering techniques so as to identify components more clearly without any ambiguity. Slicing has been applied to full-fledged software till now but here it is applied on the models. This is followed by clustering of the slices into clusters.

In this approach after the designing of the PIM, the functional dependency graph (FDG) for the whole system is drawn. The graph can be drawn with ease as the entire model providing the entities and the interaction information of the whole system have already been modeled. A component that implements a specific service in the system is isolated from the whole system. The process starts from the top and continues to go downward, looking for the top level components that characterize the desired service.

Once these functions are found, forward dependency slicing is applied taking the corresponding FDG node as the starting point. This produces slices of the model that are a part of a component and thereby have to be merged. Since a single forward dependency slice consists of all the program entities that are required for the proper operation of the service so by merging all the forward dependency slices corresponding to a particular service a component can be created. This process leads to the identification of a new component which, besides being reusable in other contexts, will typically be part of the (modular) reconstruction of the original legacy system.

But we only use the slicing technique to perform forward slices. The combination of these slices into clusters is performed using clustering technique which is as follows. Slicing technique produced the slices of the whole software which would be joined by the clustering algorithm based on the similarity between the objects. This makes use of a quantitative measure among vectors that is arranged in a matrix called proximity matrix. Two types of quantitative measures [22] are Similarity Measures, and Dissimilarity Measures. The quantitative measure used here is similarity measures. These are used to find similar pairs of objects in software. s_{xy} is called similarity coefficient between two objects x and y . Higher the similarity between objects, the higher the value of s_{xy} . For all objects x and y , a similarity measure must satisfy the following conditions [22]:

- $0 \leq s_{xy} \leq 1$
- $s_{xx} = 1$
- $s_{xy} = s_{yx}$

Slices are organized in a hierarchical structure according to the proximity matrix which in this case is similarity measure. In the hierarchical structure root represent the whole system and the leaf node represents objects. Non-leaf nodes represent the extent to which these objects are similar to each other. To combine the similar objects into a cluster merge operations are performed in a bottom-up fashion. Figure 8 represents the hybrid process of component identification. After the successful identification of components and their clustering the components are stored in the reusable components repository that can be used within the organization in other projects or can be sold off to other organizations for the development of the projects.

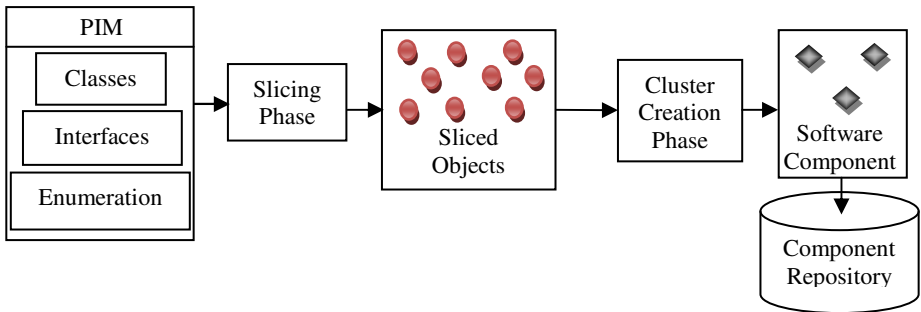


Fig. 8. Component Identification Phase

3.4 Artifact Generation Phase

This is the fourth phase of the methodology. In this phase the code artifacts are generated for the specific platforms by transforming the independent model developed in the modeling phase into platform specific models one for each of the desired language. These platform/language specific models are then further transformed to produce the code artifacts for the specific languages. The translation of PIM to PSM is carried out by a model translator that makes use of the OMG’s predefined mapping rules. In PSM the platform specific information is incorporated. From this PSM the code artifacts are generated by the code generator that makes use of the OMG’s predefined templates. The aspect code generator module generated the code for the aspect from the aspect models.

Figure 9 show the diagrammatic representation of the code generation process for some of the aspects e.g. transaction aspect and security aspect.

Figure 10 shows the diagrammatic representation of the artifact generation phase from the PSM.

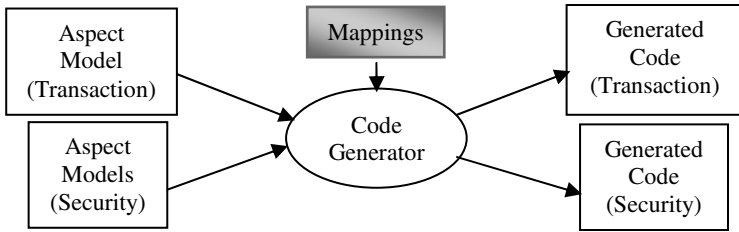


Fig. 9. Aspect Code Generation Process

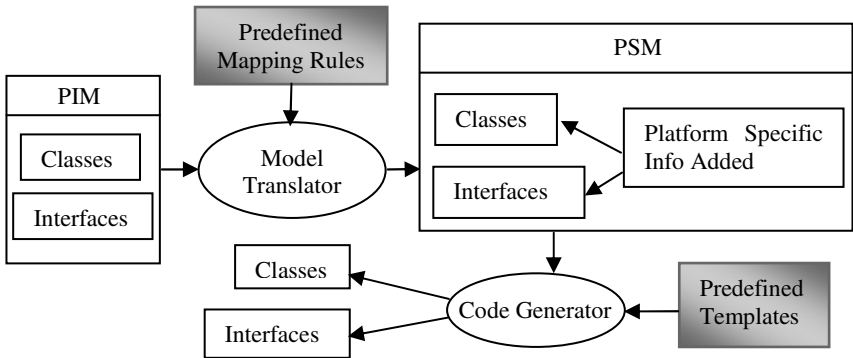


Fig. 10. Artifact Generation Phase

3.5 Integration Phase

This is the last phase of the methodology. The code generated in the code generation phase. In this phase the aspect weaver weaves the aspects to their appropriate joinpoints in the program to give a complete aspect oriented product.

Figure 11 provides a pictorial representation of the integration phase. The code generated from the PSM and aspects are supplied to the aspect weaver. The aspect weaver weaves the aspects to their corresponding joinpoints. The information about the location of weaving is interpreted from the pointcuts defined in each of the aspect. The output is the final integrated product.

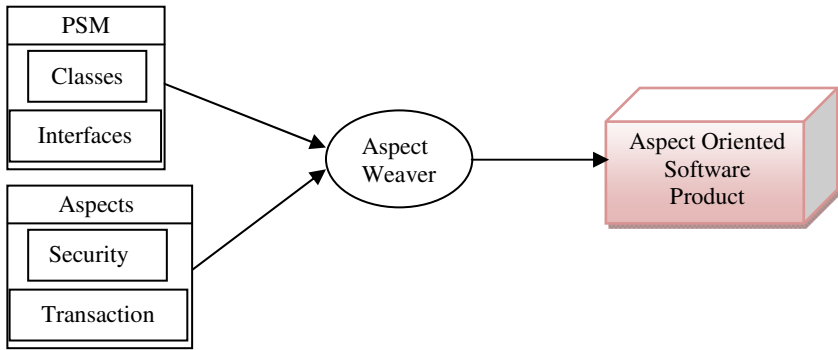


Fig. 11. Artifact Generation Phase

4 Case Study Implementation: Online Web-Forums

The significance of the proposed work is shown by implementation of an online webforum website case study. In this scenario the website allows its users to create forums, topics, posts etc. related to the different specified domains by the administrator. User can give suggestions of other domains that they want to add in the list to the administrator. The user created forums, topics or posts are published on the website within 24 hours after being duly checked by the supporting staff for any breach in the rules and policy laid down by the administrator. If any error is found the corresponding forum or post is deleted and the user is notified via e-mail. Support Staff also handles troubleshooting of various problems like password change etc.

4.1 Tools Used

For the development of the UML diagrams of the case study described above *Poseidon for UML professional edition 6.0* is used as the base tool. One of the most valuable features of Poseidon for UML and the basis for using it in the case study is its code skeleton generation technology [15]. This feature makes use of the predefined templates. The syntax of the resultant model depends on the corresponding template. The templates are predefined for many high level languages such as Java, C++, PHP, XML and HTML.

4.2 UML Diagrams

UML diagrams are the constitutional part of the proposed methodology. Whole methodology revolves around these UML diagrams as the final code artifacts are generated from these diagrams. Use-cases and the class diagrams are the main components of this methodology.

The use-case diagram depicting the functions that can be performed by the various actors is shown in figure 12. From the diagram it can be seen that there are mainly four actors that will use the website namely *Visitors* that visit the website only for exploration purpose and not for becoming the member or publish or answer a post, *Members* can create new forums/topics/posts, delete forums/topics/posts, give feedback etc, *Administrators* define the policy and rules that are to be followed, can accept or reject any visitors membership request, add or remove support staff, can suggest changes in the website to the support staff etc, Support staff are responsible for the maintenance of the website. In the diagram an actor named as generic user is added. This actor performs the actions viz. login, update details which are common for all the prominent actors' i.e. member, administrator and the support staff.

For the web-forum case study the class diagram contains all the components viz. Member, Administrator, Support Staff, Main Page, Forum Page etc. A Person class is a generalization of the Member, Administrator, and Support Staff which contains all the attributes that are common to these three classes. Figure 13 represents the class diagram for the full Discussion-Forum Website. Different Classes, Interfaces are put into their corresponding components. These components are then assembled together with the help of associations, dependencies to give a full-fledged representation of the Website.

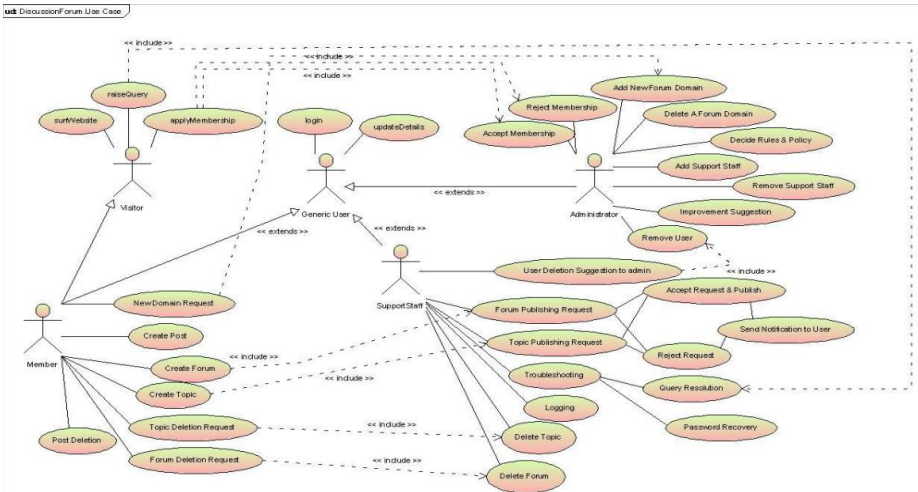


Fig. 12. Use-case for the online web-forum website

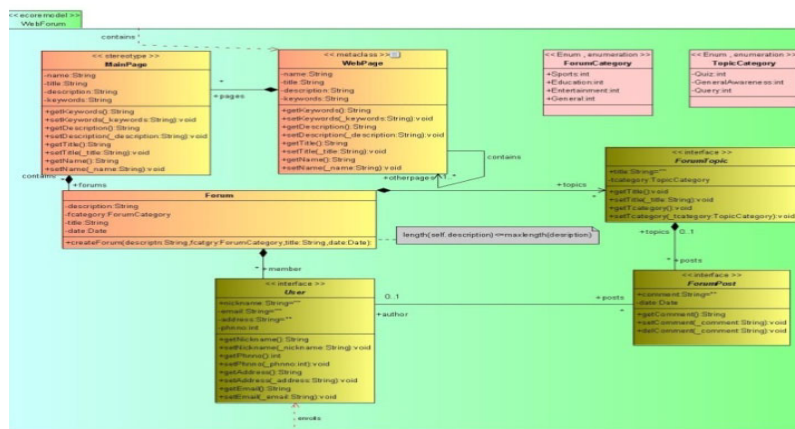


Fig. 13. Class diagram for the online web-forum website

4.3 Generated Code

The PIM developed with the help of the UML diagrams is converted to the PSM from which the code artifacts are generated. Since our case study is an online application thus the code artifact presented here is in Java and PHP as both of these languages are widely used for developing websites like the one considered in this paper.

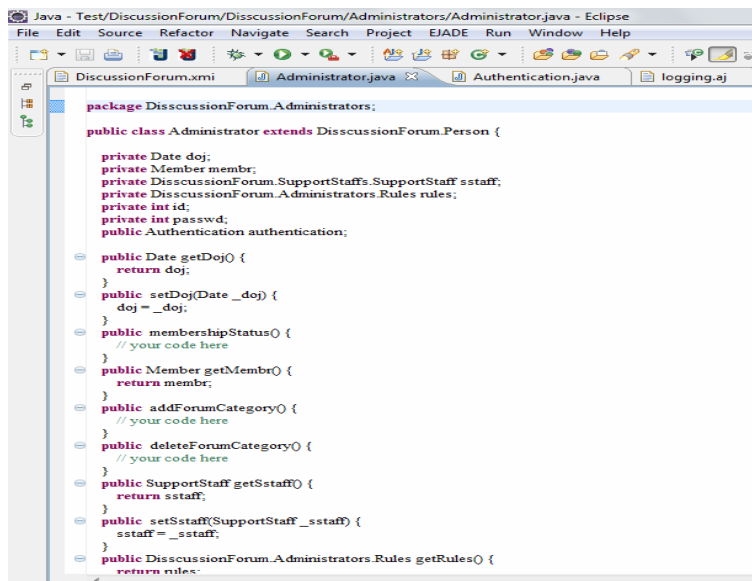
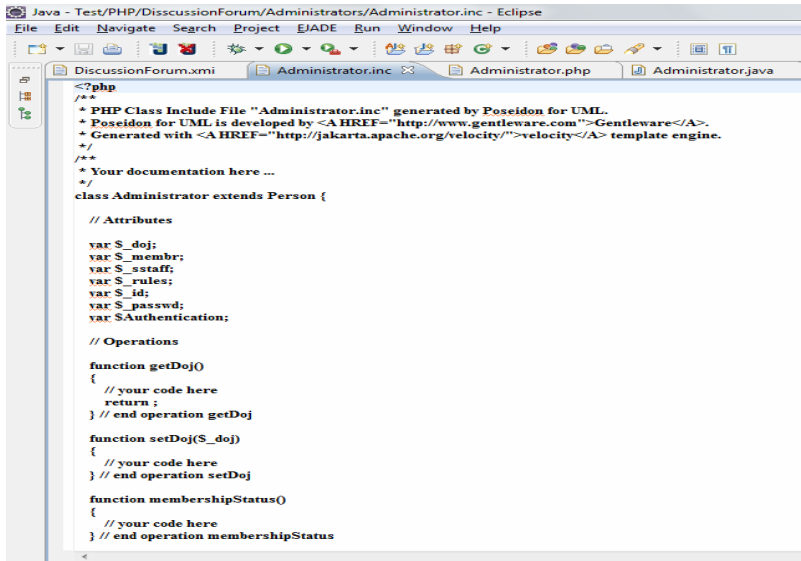


Fig. 14. Code artifact for the administrator class in Java



```

<?php
/**
 * PHP Class Include File "Administrator.inc" generated by Poseidon for UML.
 * Poseidon for UML is developed by <A HREF="http://www.gentleware.com">Gentleware</A>.
 * Generated with <A HREF="http://jakarta.apache.org/velocity/">velocity</A> template engine.
 */
/**
 * Your documentation here ...
 */
class Administrator extends Person {
    // Attributes
    var $S_doj;
    var $S_membri;
    var $S_staff;
    var $S_rules;
    var $S_id;
    var $S_passwd;
    var $SAuthentication;

    // Operations
    function getDoj()
    {
        // your code here
        return ;
    } // end operation getDoj

    function setDoj($S_doj)
    {
        // your code here
    } // end operation setDoj

    function membershipStatus()
    {
        // your code here
    } // end operation membershipStatus
}

```

Fig. 15. Code artifact for the administrator class in PHP

Figure 14 and 15 shows the code generated from the models in Java and PHP respectively for the administrator class.

5 Results and Significance

The design and implementation carried out for the proposed methodology demonstrates numerous advantages that are gained over the other prevailing approaches. The advantages that are gained by the proposed methodology are depicted as follows.

Ease of Interconversion. Since the different specific models are derived from a single PIM thus it acts as a bridge between different PSM's enabling the information how the element in one PSM relates to the other element.

Aspect Modeling. Identification and designing of aspects is done as a separate process. They are modeled separately from the business code at the initial level and are continued accordingly i.e. from PIM to PSM aspects are also modeled in same manner.

Developer Overhead Reduction. Developers have to emphasize on the coding of the functionality and not the basic structure as it is generated automatically from the model designed.

Early Error Detection. Errors are detected and rectified as early as during the design phase itself which otherwise would have been caught at testing phase and may have induced further errors during that phase.

Component Reconfiguration. After the analysis of the accumulated reuse data, the components can be reconfigured so as to make them more robust and fitting for the practical reuse.

Reusability of the Models. Since the PIM developed are stored in a PIM repository, thus they can be used as a base model to create new applications with requirements same as that of the PIM to be used.

Reusability of the Aspects. Design aspects are also stored in an aspect repository so that the aspect model can be used in other applications with the need to develop them.

Reusability of the Components. As initially independent components have been identified during modeling thus those would correspond to independent components in coding that can be reused in other products of similar type. This is represented in figure 12 where a component from product 1 is being reused in product 2.

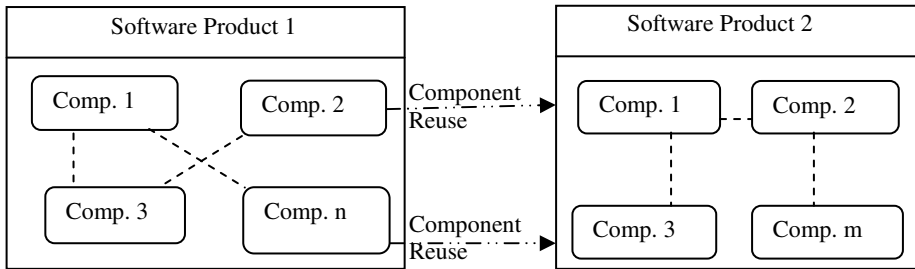


Fig. 12. Reuse of the components in other software

6 Conclusion

Designing and transforming of model has been the core functionality of MDA. The methodology proposed integrates the concept of aspects and software reuse in MDA. Models are developed and later transformed to generate the code artifacts for the specific platform. Thus developer only lays emphasis on writing the code for the specific functionality and thereby reduces a considerable amount of burden for the developer. The division of the product into the components enables the concept of software reuse. Along with reuse of the components the PIM developed can be reused in other products of same specification and structure. Aspects are modeled separately from the business logic thereby eliminating the problem of scattering and tangling. Thus the major concepts in software development viz. reusability and aspects get incorporated into the MDA approach to make it more robust in nature.

References

1. Object Management Group: MDA Guide Version 1.0.1. (2011)
<http://www.omg.org/mda/> (Last accessed February 21)
2. Jacobson, I., Christerson, M., Jonsson, P., Övergaard, G.: Object-Oriented Engineering, ACM Press. Addison Wesley, Reading (1992)

3. IEEE-SA Standards Board: IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. IEEE Std 1471-2000, pp.1–23 (2000)
4. Pour, G.: Moving toward Component-Based Software Development Approach. IEEE J. Technology of Object-Oriented Languages and Systems (1998)
5. Wu, Y., Offutt, J.: Maintaining Evolving Component-Based Software with UML. In: Proc. of the 7th IEEE European Conference on Software Maintenance and Reengineering (2003)
6. Cai, X., et al.: Component-Based Software Engineering: Technologies, Development Frameworks, and Quality Assurance Schemes. In: Proc. of the 7th IEEE Asia-Pacific Software Engineering Conference, pp. 372–379 (2000)
7. Jin, X.: Applying Model Driven Architecture approach to Model Role Based Access Control System. Master Thesis. University of Ottawa, Ottawa, Ontario, Canada (2006)
8. Elrad, T., et al.: Special Issue on Aspect-Oriented Programming. Communications of the ACM 44(10), 29–32 (2001)
9. Fuentes-Fernández, L., Vallecillo-Moreno, A.: An Introduction to UML Profiles. Journal of Informatics Professional 5(2), 6–13 (2004)
10. Wang, Z., Xu, X., Zhan, D.: A Survey of Business Component Identification Methods and Related Techniques. International Journal of Information Technology 2(4), 229–238 (2005)
11. Object Management Group, UML Resource Page (2011), <http://www.omg.org/uml/> (Last accessed February 21)
12. Fuentes-Fernández, L., Vallecillo-Moreno, A.: An Introduction to UML Profiles. Journal of Informatics Professional 5, 6–13 (2004)
13. Bast, W., Kleppe, A., Warmer, J.: MDA Explained: The Model Driven Architecture: Practice and Promise. Addison-Wesley Publication, Reading
14. Rodrigues, N. F., Barbosa, L.S.: Component Identification through Program Slicing. Electronic Notes in Theoretical Computer Science (2005)
15. Boger, M., Graham, E., Köster, M.: Poseidon for UML (2011), <http://www.gentleware.com/fileadmin/media/pdfs/userguides/PoseidonUsersGuide.pdf> (Last accessed February 5)
16. Fanchao, M., Dechen, Z., Xiaofei, X.: Identifying Business Components from Business Model: A Method Based on Feature Matching. In: International Conference «e-Management and Business Intelligence», pp. 1–9 (2007)
17. Höppner, F., Klawonn, F., Kruse, R.: Fuzzy Cluster Analysis: Methods for Classification, Data Analysis, and Image Recognition. Wiley, New York (1999)
18. Fan-Chao, M., Den-Chen, Z., Xiao-Fei, X.: Business Component Identification of Enterprise Information System: A hierarchical clustering method. In: Proc. of the IEEE International Conference on e-Business Engineering, pp. 473–480 (2005)
19. Lung, C.H., Zaman, M., Nandi, A.: Applications of Clustering Techniques to Software Partitioning, Recovery and Restructuring. Journal of System Software 73(2), 227–244 (2004)
20. Chesman, J., Daniels, J.: UML Components. A Simple Process for Specifying Component-Based Software. Addison-Wesley, Upper Saddle River (2001)
21. Aßmann, U.: Model Driven Architecture (MDA) and Component-Based Software Development (CBSD). In: Xootic Magazine Feb.07 Edition, pp 5–7 (2007)
22. Shahmohammadia, G.R., Jalilia, S., Hasheminejada, S.M.H.: Identification of System Software Components Using Clustering Approach. Journal of Object Technology 9(6), 77–98 (2010)

23. Zhang, J., Chen, Y., Li, H., Liu, G.: Research on Aspect-Oriented Modeling in the Framework of MDA. In: Proc. of the 2nd IEEE International Conference on Computer Science and Information Technology (ICCSIT), pp. 108–111 (2009)
24. Laddad, R.: AspectJ in Action, 2nd edn. Manning Publication (2009)
25. Simmonds, D., et al.: An Aspect Oriented Model Driven Framework. In: Proc. of the 9th IEEE International EDOC Enterprise Computing Conference (2005)
26. Simmonds, D.M., Raghu Reddy, Y., Song, E., Grant, E.: A Comparison of Aspect-Oriented Approaches to Model Driven Engineering. In: Proc. of Conference on Software Engineering Research and Practice, pp. 327–333 (2009)
27. Simmonds, D.M.: Aspect-oriented Approaches to Model Driven Engineering. In: Proc. of the International Conference on Software Research and Practice, Las Vegas, Nevada, USA (2008)
28. Reina, A.M., Toress, J., Toro, M.: Towards developing generic solutions with aspects. In: Proc. of the Workshop in Aspect Oriented Modeling Held in Conjunction with UML (2004)
29. Jacobson, I.: Case for Aspects - Part I. Software Development Magazine, 32–37 (2003)
30. Jacobson, I.: Case for Aspects - Part II. Software Development Magazine, 42–48 (2003)
31. Kulkarni, V., Reddy, S.: Separation of Concerns in Model driven Development. IEEE Software 20(5), 64–69 (2003)
32. Song, E., Reddy, R., France, R., Ray, I., Georg, G., Alexander, R.: Verifying Access Control Properties using Aspect Oriented Modeling. In: 10th ACM Symposium on Access Control Models and Technologies (SACMAT), Scandic Hasselbacken, Stockholm (2005)
33. Mellor, S.: A Framework for Aspect-Oriented Modeling. In: 4th AOM Workshop at UML2003, San Francisco, CA (2003).

Application of 80/20 Rule in Software Engineering Rapid Application Development (RAD) Model

Muhammad Rizwan¹ and Muzaffar Iqbal²

¹ University Institute of Information Technology (UIIT) – Pir Mehr Ali Shah Arid Agriculture
University Rawalpindi, Pakistan
rizwan.uaar@gmail.com

² Federal Urdu University of Arts, Science & Technology Islamabad, Pakistan
muzaffar.iqbal123@yahoo.com

Abstract. Efficiency of software process models is exceedingly necessary for the software process model improvement. The software project managers can ensure efficient implementation of software process models. As a result, the efficient software Rapid Application Development (RAD) model makes the software process efficient and ultimately the efficient software project management at work. The historic application of 80/20 method in socio-economic field and in the field of software project management was the inspiration of this work. All this led us to work on the application of 80/20 rule in software engineering Rapid Application Development (RAD). The ultimate result of our research work is the improvement of RAD model by focusing on fewer activities which can give 80 percent of the overall productivity of the software process at work. Our work is actually facilitating software engineers by focusing only on the critical activities and not to devastate their time and energies on the activities which are just producing a small portion of the overall outcome.

Keywords: Pareto's principle, 80/20 Rule, software engineering process model, Rapid Application Development (RAD), software project management.

1 Introduction

The Pareto Principle or Law of Vital Few was given by Vilfredo Pareto. It was observed by Vilfredo Pareto (1848-1923), the Italian economist in 19th century in Italy, that 20 percent of the population owned 80 percent of the usable land (Pareto, 1935). The same distribution found by Pareto in other economical and natural processes. As a general rule, this finding was formulated by Sombart (1967) as: "in any arbitrary set of elements, that try to achieve something a subset small in numbers will have the biggest effect".

According to Walter Maner [15] RAD Model reduces the development time and reusability of components help to speed up development. Throughout the software development lifecycle (SDLC), if commitment is lacking, RAD will fail. So, if work-task part (phases) of the RAD model is improved, then software process will improve and ultimately, software project management will improve. This improvement in

work-task part can be achieved to assist software engineers at work, with the improved model to help them achieve high productivity (performance) in less time and effort. This research work has applied 80/20 rule on the Rapid Application Development (RAD) model to achieve the improved model in term of improved work-task part.

We have already applied 80/20 rule successfully on waterfall process model [14]. This current work is extension of our earlier work in order to use RAD model more successfully and efficiently in lesser time and effort.

The rest of the paper is organized as below. Section 2 covers the literature review and related work. Section 3 describes how this research was planned and executed. Section 4 shows the results, and discussion of this research. Sections 5 and 6 show the conclusion, case studies and future work respectively.

2 Literature Review and Related Work

The application of 80/20 rule Pareto Principle or Law of Vital Few in the software engineering process models is implemented in our research work. But the rule was actually suggested for social and economical fields, as it states that: '80 percent of outcome can be gained through 20 percent of resources'.

According to Simon (2007), in social and economic fields, 80 percent of process defects arise from 20 percent of the process issues, 20 percent of your sales force produces 80 percent of your company revenues, 80 percent of the delays in schedule arise from 20 percent of the possible causes of the delays, and 80 percent of the customer complaints arise from 20 percent of your products or services.

According to Pareto (1935), 80 percent of the consequences come from 20 percent causes in economics. Pareto's 80/20 rule can be used in these situations, like 20 percent of customers generate 80 percent of turnover, 20 percent of products make 80 percent of turnover, 20 percent of possibilities to make faults in production are responsible for 80 percent of product defects, 80 percent of the decisions are made in 20 percent of the time in a meeting, 20 percent of products make 80 percent of profit, 20 percent of employees account for 80 percent of the time absent, 80 percent of results are achievable in 20 percent of working time if strategic time planning is used, the best 80 percent of sellers are responsible for 80 percent of the profit of a firm, 20 percent of the goods in a stock sum up to 80 percent of the stock worth, 80 percent of the requests for stocked articles are on only 20 percent of the goods, and 80 percent of the costs or losses of a business are caused by 20 percent of the problems (Pareto, 1935).

According to Ultsch [1], in any field 20 percent of the total efforts give you much or 80 percent of the total output, which is the basic essence of 80/20 Rule. The author Ultsch (2002) has used 80/20 rule in the ABC-analysis, which is commonly used for the optimization of business and projects. According to author, if projects having subdivision can lead to optimized cost, in turn it can yield much output as a whole by less cost (in the form of effort). This leads one to optimize efforts and give out high performance.

According to Koch [2] there are ways that how 80/20 Rule can be applied in the real life in any field, as it is being used in economic and social fields. The author

offers one to think any other use of the 80/20 rule in any field of your work. The author has motivated others as that if we focus on our important and valuable activities, then we can have much time to do other activities, as your 20 percent focused effort has given you the 80 percent output of the whole development effort.

The most related research work is [14], in which we have successfully applied 80/20 rule on waterfall process model. This work has proved that some of the activities of Waterfall process model are more important to focus than others, being more productive at work. So, if we focus only on the activities which are producing most of the outcome, then a lot of software development and management time can be saved.

3 Research Implementation

Historical descriptive (survey based) research method (Kuh et al. (2005), Patton (1990)), of research was adopted to conduct this study through Rapid Application Development (RAD) Model evaluation questionnaire.

One structured interview was prepared to find out the usability of software process models being used in different software houses according to the nature of software projects developed in organic mode of COCOMO model using the programming languages (C#, Java, Visual basic, and ASP) through the structured interview.

Next step was to prepare a detailed Rapid Application Development (RAD) model evaluation questionnaire to evaluate the RAD model being used for the software projects developed in organic mode of COCOMO model using the programming languages (C#, Java, Visual basic, and ASP). To achieve this step, we jotted down all the possible activities in four phases (stages) of the RAD model. All the contents for the RAD model evaluation questionnaire were gathered from different sources including [3][4] [5][6][7][8][12][13], and other related web sources.

The contents were organized in the form of the evaluation questionnaire (Phases→Activities, Activities→Sub-activities).

About twenty-four software houses having CMMI Level 2 & 3 were visited personally by the researchers. Eighty software practitioners (software engineers, software project managers and software development members) of visited software houses were selected randomly as sample from the population for the filling of questionnaire on the basis of their sound experience being the active part of the software industry using the RAD model. The criterion for selection of software practitioners and experts was that they should be experienced people of their software organizations having at least 5 years experience, and they have implemented 8-10 quality software projects using the RAD model, and observed the impact, usability and effectiveness of different activities for the productivity RAD model. The RAD model questionnaire was distributed to the selected software practitioners.

4 Results and Discussions

On the basis of available literature [3][4][5][6][7][8][12][13], we found 115 activities of the Rapid Application Development (RAD) model, and designed in the form of

Rapid Application Development (RAD) model evaluation questionnaire. The Rapid Application Development (RAD) model evaluation questionnaire was evaluated by the software practitioners to apply 80/20 rule according to the firm and experienced observations of the software practitioners using the Rapid Application Development (RAD) Model for software(s) development under the organic mode [10] of COCOMO model using the programming languages (C#, Java, Visual basic, and ASP). This led us to focus on the 70 activities to give us 70 to 80 percent of the whole productivity through the reduction of the effort and increasing the performance of the model. Also, there were 45 activities which could be ignored, eliminated or delegated, as the software practitioners found these activities giving less or 20 percent of the overall productivity of the RAD process at work. This research result is to facilitate the software project managers and the software professionals who face the problem of making the software process model efficient for the software development throughout the Software Development Life Cycle (SDLC).

a. RAD Model Activities to be ignored:

On the basis of our results, the following activities can be ignored or eliminated by software practitioners while implementing the Rapid Application Development (RAD) Model for software developed as observed contributing 20 percent of overall productivity and efficiency of RAD model:

1. Requirements Planning & Specification Phase [10][12]:

i. In the ‘Planning of User Requirements’ activity:

- Gather the software requirements using Questionnaires
- Identify the user/client software requirements as Non-Functional (External) Requirements
- Review the proposed system areas instantaneously
- Prioritize the proposed system areas instantly
- Finalize the proposed system areas instantaneously

ii. In the ‘Defining User Requirements’ activity:

- Justify the cost feasibility of software system through COCOMO Model and Wideband Delphi Process
- Justify the behavioral feasibility of software system through Bench-Marks and Perspectives
- Justify the technical feasibility of software system through Functional Points and COCOMO Model

iii. In the ‘Modeling of Information Flow’ activity:

- Determine, information movement
- Determine, information processing source

2. User Design Phase [10][12]:

i. In the ‘Modeling of Data’ activity:

- Define the user design

ii. In the ‘Detailed Analysis of Software System’ activity:

- Designing of the system’s procedures:
 - Designing Procedure of Project Management
 - Designing Procedure of System Analysis

- Designing Procedure of Software Design
- Designing Procedure of Software Programming
- Designing Procedure of Deployment Implementation
- Develop the initial layout screens of software system
 - Develop external layout screens
 - Build prototypes of critical software procedures, by creating DFD – Level 1

iii. In the ‘Modeling of Data Flow’ activity:

- Define the data object composition
- Define the object attributes
- Identify location of the data objects
- Define the relationship between the objects and the objects' transformation processes

3. Construction Phase [10][12]:

i. In the ‘Coding , Implementation and Testing’ activity:

- Creating operating documentations:
 - Creating Programming Guidelines
 - Creating Screen Shots
- Creating the interfaces:
 - Internal Interfaces
- Creating the databases:
 - Creating real time Databases
 - Creating distributed databases
 - Usage of the language to create the database Oracle 8i
 - Usage of the language to create the database Oracle 9i
 - Usage of the language to create the database Oracle 10g
 - Usage of the language to create the database MS Access
- Integrating the internal interfaces of software
- Performing planning for System Testing
- Developing the formal test plan using 'Clean-room process'
- Developing (refining) the formal test plan through Walkthrough
- Identifying and documenting test requirements
- Executing the unit tests of software system
- Executing the System Testing as a whole
- Executing system testing directly with users
- Executing system testing using Time boxing
- Executing system testing along with parallel software development

ii. In the ‘Using 4th Generation techniques for software development’ activity:

- Creating reusable components through automated tools

4. Turnover Phase [10][12]:

i. In the 'Acceptance testing of Client and Transition' activity:

- Performing planning of Developed Software installation at Client Side
- Organizing & Delivering the training to integrate the system components
- Identifying the possible upcoming software enhancements
- Identifying the likely support for the client

b. RAD Model Activities which must not be ignored:

On basis of our results, the following activities which must be followed by software practitioners while implementing the Rapid Application Development (RAD) Model for software development as observed giving 80 percent contribution in overall productivity of RAD model:

1. Requirements Planning & Specification Phase [10][12]:

i. In the 'Planning of Requirements' activity:

- Define planning of requirements
- Gather the user requirements (through Interviews, Bench-Marks & Perspectives)
- Identify the user/client Functional Requirements

ii. In the 'Defining User Requirements' activity:

- Developing the use cases diagram/model
- Estimate the complexity of software system to refine the system's scope

iii. In the 'Modeling of Information Flow' activity:

- Determine the driving source (information) of the business process
- Determine the output information after successful business process execution
- Develop the use cases & Use Case Diagram (UCD)

2. User Design Phase [10][12]:

i. In the 'Modeling of Data' activity:

- Defining information flow in detail for refining it into a set of data objects
- Determine the features (attributes) of each object (entity)
- Define the interactions (relationships) between the defined objects
- Develop ERD of the defined software system at team level
- Develop the Activity Diagram of each use cases

ii. In the 'System Detailed Analysis' activity:

- Designing the system procedures:
 - Designing 'Procedure of Project Definition'
 - Designing 'Procedure of Software Design'
 - Designing and developing the preliminary layouts of the screens
 - Designing and building prototypes of important events, by:
 - Developing DFD - Level 0 of the software system
- Integrate the functions of the software system with the objects
- Documenting the processing description for manipulating (adding, modifying, deleting, retrieving) data objects, by:
 - Developing Sequence Diagram (SD) of software system
 - Developing Class Diagram (CD) of software system

iii. In the ‘Modeling of Data Flow’ activity:

- Identify the most important objects of the system
- Identify the relationships between objects of software system

3. Construction Phase [10][12]:**i. In the ‘Coding and Testing’ activity:**

- Produce test data for testing of software system
- Develop source code of software system
- Develop object code of software system
- Developing user manual of developed software system
- Performing planning for Integration of software components (modules) by:
 - Performing finalization of the software design
 - Developing the External Interfaces of software system
- Developing the databases like Relational Database:
 - Usage of the language to create the database SQL
- Performing integration of external interfaces of software system
- Performing planning of testing (for Unit Testing and System Integration Testing)
- Developing and documenting the formal test plan
- Executing the test cases
- Performing direct implementation
- Executing testing of whole system
- Monitoring software implementation and finalization progresses to complete each task quickly
- Performing the review the documentation on any software change
- Do software iteration (after each software change)

ii. In the ‘Using 4th Generation techniques for software development’ activity:

- Reusing existing software components through automated tools

4. Turnover Phase [10][12]:**i. In the ‘Acceptance testing and turnover (transition)’ activity:**

- Developing the deployment diagram of software system
- Distributing the software to the client
- Installing software at client site
- Performing actual operations of the software system
- User acceptance for the software system in operational environment (User acceptance testing)
- Performing planning of the software training for the client (end-users)
- Delivering the training to operate the software system and for maintenance of the software system
- Test the new components enhancement (if any change occurs, and fixed)
- Conform the SRS (Software Requirements Specification) and software design

The contents were organized in the form of the evaluation questionnaire (Phases→Activities, Activities→Sub-activities), and evaluation parameters were provided to rate each activity and/or sub-activity on the basis of its impact on the productivity of the Rapid Application Development (RAD) model performance. The

survey response (facts) against each activity of each phase of Rapid Application Development (RAD) Model was actually weights assigned by Software Practitioners as function point for each activity or sub-activity of the different phases of the Rapid Application Development (RAD) process model.

The survey response (facts) of the software practitioners was first presented for evaluation, and then 80/20 rule was applied on the evaluated RAD model evaluation questionnaire, to identify the list of activities which were observed by software practitioners as giving 80 percent of the overall productivity. This activity was performed to extract improved form of the RAD model after this evaluation by software practitioners by applying 80/20 rule on the Rapid Application Development (RAD) model. In our research, we preferred to use computation models for extracting our desired results, because computation based methods are proven to be more accurate and effective. We have preferred to get industrial practices and software industrial practitioners' experienced-based facts as input for the implementation of our research.

In the Rapid Application Development (RAD) Model Evaluation Questionnaire, each phase of the Rapid Application Development (RAD) Model was divided into activities and activities were subdivided into the possible tasks. Within each phase, software practitioner(s) evaluated each activity with respect to other in terms of its potential to give high output at work, as observed at work while implementing the Rapid Application Development (RAD) Model for software(s) development under organic mode of COCOMO model using the programming languages (C#, Java, Visual basic, and ASP). The Evaluation questionnaire contained evaluation parameters against each activity/or task of each phase of the Rapid Application Development (RAD) Model i.e. S.A. =5 if giving much or 80 percent output relevant to other, A=4 if giving 60 percent output relevant to other, U.C. =3 if giving less than 50 percent output relevant to other, D. =2 if giving less than 40 percent output relevant to other, S.D. =1 if giving less than 20 percent relevant to other.

This was obtained on the basis of the results obtained through the software process model evaluation questionnaire for each software process model. In the questionnaire, the software professionals from the software industry evaluated the software Rapid Application Development (RAD) model, by assigning the weights according to our evaluation parameters values according to the status of any activity or task according to its criticality in terms of giving productivity overall while implementing Rapid Application Development (RAD) model for software(s) development under the COCOMO organic mode of effort and performance using the programming languages (C#, Java, Visual basic, and ASP) at work.

The analysis and estimation techniques used in our research work are COCOMO (Constructive Cost Model) [10], Function Point Analysis (FPA) [9], and QSM Function Point Programming Languages Table [11]. An analysis sheet was prepared to calculate the total raw modified function points (total Function Points (FP) [9]) before and after application of 80/20 Rule in software Rapid Application Development (RAD) process model, which is one of the software project management quality parameters of the effort and performance [9][10].

The activities which came in the region of S.A. and A. (means that those activities are very productive at work, as observed by the software practitioners) according to facts given as response in the Rapid Application Development (RAD) Model

Evaluation Questionnaire, were selected for the improved software Rapid Application Development (RAD) process model after application of 80/20 rule. Each rating against each activity or task within the phases of the software Rapid Application Development (RAD) model, were considered as raw modified function points for the whole RAD process model for the softwares developed in the COCOMO. The total raw modified function points (FP) of Rapid Application Development (RAD) Model were counted as 429 without the application of 80/20 rule on the Rapid Application Development (RAD) Model.

For the research implementation before and after application of 80/20 rule in the software Rapid Application Development (RAD) model, as shown in the Table 1 below, the total raw modified function points (FP) of Rapid Application Development (RAD) Model were counted as 275 after the application of 80/20 rule on the Rapid Application Development (RAD) Model.

Using the above counted values of raw modified function points (FP) of the whole Rapid Application Development (RAD) model before and after application of 80/20 rule, we counted the values for the rest of eight software project management quality parameters of the effort and performance i.e. total environmental influence factor (N) [9], Complexity adjustment factor (CAF) [9], Adjusted Function Points (AFP) [9], KLOC[9][10], Effort (E) [10], TDEV [10], average staff estimate (SS) [10], PRODUCTIVITY (P) [10].

A separate questionnaire was prepared and distributed to the software practitioners to get the values of 14 environmental influence factors for the software projects developed in the organic mode of COCOMO model using the programming languages (C#, Java, Visual basic, and ASP). With the positive response (facts) of the software practitioners, the total environmental non-functional influence factor (N) was counted and used for analysis of the research results using COCOMO (Constructive Cost Model) [10] and Function Point Analysis (FPA) [9].

The function point analysis environmental influence factors of software development rated were Data Communications [9], Distributed Computing [9], Performance Requirements [9], Constrained Configuration [9], Transaction Rate [9], Online Inquiry and/ or Entry [9], End-User Efficiency [9], Online Update [9], Complex Processing [9], Reusability [9], Ease of Conversion/ Install [9], Ease of Operation [9], Used at Multiple Sites [9], and Potential for Function Change [9]. Then, software practitioners rated each factor as factor rating (fn), then average rating against each factor was counted.

Then, for the research implementation before and after application of 80/20 rule in the software Rapid Application Development (RAD) model, as shown in the table 1 below, the total environmental influence factor (N) [9] was counted by the sum of the factor rating (fn) [9], as feedback (facts) about the effect of function point analysis environmental factors while Software Development Life Cycle (SDLC), by the software project managers. Then, using total of N for results before and after application of 80/20 Rule in Software Rapid Application Development (RAD) model, we calculated CAF (Complexity Adjustment Factor) [9], to use in the implementation of COCOMO (Constructive Cost Model) for the generalized proof for the validation of the results of the application of 80/20 rule in the software Rapid Application Development (RAD) process model, using COCOMO (Constructive Cost-estimation Model) [10] and FPA (Function Point Analysis) [9].

For the research implementation before and after application of 80/20 rule in the software Rapid Application Development (RAD) model, taken from the actual results of the Function Point Analysis Environmental Factors Rating Questionnaire, total function point analysis environmental influence factors of software development (N) was used to estimate the complexity adjustment factor (CAF) [10] using formula, $CAF = 0.65 + (0.01 * N)$, where 0.65, and 0.01 are constants., which is shown in the table 1 below.

Then, for the research implementation before and after application of 80/20 rule in the software Rapid Application Development (RAD) model, using the raw function points (FP) [10] and CAF (complexity adjustment factor), as shown in the table 1, we estimated Adjusted Function Points (AFP) [10] using formula: $AFP = (\text{raw FP}) * CAF$, where raw FP are the raw function points for entire Rapid Application Development (RAD) Model, and CAF is the complexity adjustment factor calculated in above step.

Then, for the software projects developed in programming languages (C#, Java, Visual Basic, and ASP) under organic mode [9], we used the result of AFP as input to convert it into LOC (Lines of Code) [9]. Using formula: $LOC = AFP * LOC / AFP$, where LOC / AFP [9] is the rate of Lines of code to the Adjusted Function Points, estimated above. This rate (LOC / AFP) [11] has been taken for each of the software development languages i.e. for C# as 59, for Java as 60, for Visual Basic as 50, and for ASP as 69.

Then, for the research implementation before and after application of 80/20 rule in the software Rapid Application Development (RAD) model, as we only used KLOC (Kilo Lines of Code) [9][10] as size for estimating the effort using COCOMO Model, so we converted this LOC into KLOC, by dividing it by 1000, to get KLOC of LOC, as shown in the table 1 below.

Then, we applied the COCOMO Model [10] as a simulation of further implementation of the research results as evidence. This application was consisted of calculating the COCOMO Effort for Basic Organic Mode. For the research implementation before and after application of 80/20 rule in the software Rapid Application Development (RAD) model, for the software projects developed in programming languages (C#, Java, Visual Basic, and ASP) under organic mode [10], as shown in the table 1 below, COCOMO Effort (E) [10] was estimated by using formula: $\text{Effort (E)} = a * (\text{Size})^b$, where a and b are the constants derived from regression analysis (depends on the project). Also, size was expressed in thousands of lines of code (KLOC) [9][10]. Then also, effort was expressed in staff-months. Effort for Organic Mode is given by formula: $E = 2.4 * (\text{Size})^{1.05}$ [10], where 2.4 and 1.05 are the constants for the organic mode of COCOMO effort and performance.

Then, for the research implementation before and after application of 80/20 rule in the software Rapid Application Development (RAD) model, as shown in the table 1 below, for the software projects developed in programming languages (C#, Java, Visual Basic, and ASP) under organic mode [10], Basic COCOMO Project Duration estimate (TDEV) [10] was estimated by using formula: $TDEV = a * (E)^b$, where a and b are constants derived from regression analysis (depends on the project). Also, effort for the software projects in organic mode of effort and performance was counted in staff-months [10]. Next to this measurements implementation or implementation of COCOMO and FPA in our research, development time for Organic Mode of COCOMO Effort and Performance was given by formula: $TDEV = 2.5 * (E)^{0.38}$ [10],

where 2.5, and 0.38 are the constants for the software projects developed in organic mode of effort and performance. This project measurement time was expressed in months [10].

Then, for the research implementation before and after application of 80/20 rule in the software Rapid Application Development (RAD) model, as shown in the table 1 below, for the software projects developed in programming languages (C#, Java, Visual Basic, and ASP) under organic mode [10] of COCOMO Effort and Performance, Basic COCOMO average Staff estimate (SS) [10] was estimated by using formula: Average Staff: $SS = \text{Effort} / \text{TDEV}$ [10], where Effort is the estimated effort in person-months; and TDEV is the development time in months for the software projects of low complexity, and developed in stable environment.

In the last, for the research implementation before and after application of 80/20 rule in the software Rapid Application Development (RAD) model, as shown in the table 1 below, for the software projects developed in programming languages (C#, Java, Visual Basic, and ASP) under organic mode [10] of effort and performance, Basic COCOMO Average productivity (P) [10] was estimated using formula, i.e.: Productivity: $P = \text{Size} / \text{TDEV}$ [10], where size is the estimated volume, and TDEV is the estimated time duration of the software project(s) developed in low complexity, and in stable environment according to the COCOMO organic mode of effort and performance. All the implementation of Function Point Analysis (FPA) and COCOMO (Constructive Cost Model) was performed before and after application of 80/20 rule in the Rapid Application Development (RAD) Model.

The calculated values of the software project management quality parameters of the effort and performance [9][10] were compared in the table below to show the improvement in the work-task part of software Rapid Application Development (RAD) Model resulted only after application of 80/20 rule on the software Rapid Application Development (RAD) model.

On the basis of analysis of the study, the conclusions were drawn and recommendations were made to develop the strategies for improving further the work-task part of the software project management i.e. improving the software process models, being a software project manager.

As shown in the table 1 below, the measured values of software project management quality parameters [9][10] of the effort and performance (i.e. FP (total function points of the phases of the software process model), N (total environmental non-functional influence factors), CAF (complexity adjustment factors), AFP (adjusted functional points), KLOC (kilo lines of code), E (effort) unit of measurement is persons- month, TDEV (project time duration) unit of measurement is persons- month, SS (average staff) unit of measurement is persons) after the application of 80/20 rule in the Rapid Application Development (RAD) model is reduced as compared to the measured values of these parameters before application of 80/20 rule in the Rapid Application Development (RAD) model.

The improvement in terms of reductions in measured values of software project management quality parameters [9] [10] of the performance (i.e. P (Productivity)) is increased and performance has increased after the application of 80/20 Rule in the Rapid Application Development (RAD) model. So, the software process has improved and ultimately software project management has improved with the provision of this valuable assistance for software project management at work).

Software project managers have applied and focused on our given improved Rapid Application Development (RAD) model (list of activities giving much (80 percent of productivity of whole model performance). The same improved results have been obtained by the software practitioners at work during development of different software projects developed in organic mode using programming languages (C#, Java, VB, and ASP) using our improved Rapid Application Development (RAD) model.

1. FP (Total Function Points of the phases of the Rapid Application Development (RAD) model)
2. N (Total Environmental Non-functional Influence Factors) [9][10]
3. CAF (Complexity Adjustment Factors) [9][10]
4. AFP (Adjusted Functional Points) [9][10]
5. KLOC (Kilo Lines of Code) [9][10]
6. E (Effort) - Unit of measurement is Persons- Month [9][10]
7. TDEV (Project Time Duration) - Unit of measurement is Persons-Month (PM) [9][10]
8. SS (Average Staff) - Unit of measurement is Persons [9][10]
9. P (Productivity) [9][10]

Table 1. Comparison Results of the Software Project Management Quality Parameters of the effort and performance between Rapid Application Development (RAD) Model with Application of 80/20 Rule and Without Application of 80/20 Rule

FP¹	Before application of 80/20 rule in Rapid Application Development (RAD) Model	429	429	429	429
	After application of 80/20 rule in Rapid Application Development (RAD) Model	275	275	275	275
N²	Before application of 80/20 rule in Rapid Application Development (RAD) Model	48.8	48.8	48.8	48.8
	After application of 80/20 rule in Rapid Application Development (RAD) Model	48.8	48.8	48.8	48.8
CAF³	Before application of 80/20 rule in Rapid Application Development (RAD) Model	1.138	1.138	1.138	1.138
	After application of 80/20 rule in Rapid Application Development (RAD) Model	1.138	1.138	1.138	1.138
AFP⁴	Before application of 80/20 rule in Rapid Application Development (RAD) Model	488.2	488.2	488.2	488.2
	After application of 80/20 rule in Rapid Application Development (RAD) Model	312.9	312.9	312.9	312.9
KLOC⁵	Before application of 80/20 rule in Rapid Application Development (RAD) Model	28.80	29.29	24.41	33.68

Table 1.(continued)

	After application of 80/20 rule in Rapid Application Development (RAD) Model	18.46	18.77	15.65	21.59
E⁶	Before application of 80/20 rule in Rapid Application Development (RAD) Model	81.77	83.23	68.73	96.39
	After application of 80/20 rule in Rapid Application Development (RAD) Model	51.26	52.18	43.09	60.43
TDEV⁷	Before application of 80/20 rule in Rapid Application Development (RAD) Model	13.33	13.42	12.47	14.18
	After application of 80/20 rule in Rapid Application Development (RAD) Model	11.16	11.23	10.45	11.87
SS⁸	Before application of 80/20 rule in Rapid Application Development (RAD) Model	6.136	6.203	5.509	6.794
	After application of 80/20 rule in Rapid Application Development (RAD) Model	4.593	4.644	4.124	5.086
P⁹	Before application of 80/20 rule in Rapid Application Development (RAD) Model	352.2	351.9	355.2	349.5
	After application of 80/20 rule in Rapid Application Development (RAD) Model	360.1	359.8	363.1	357.3
Languages to be used for software development using the organic mode of the effort and performance according to COCOMO Model		C#	JAVA	VISUAL BASIC	ASP

These results are showing that if software practitioners use our improved Software Rapid Application Development (RAD) model while developing software applications and systems developed under organic mode of COCOMO model, then they will get improvement in the software project management quality parameters like reduction in the KLOC (Kilo Lines of Code) [9][10], E (Effort) - Unit of measurement is Persons- Month [9][10], TDEV (Project Time Duration) - Unit of measurement is Persons-Month (PM) [9][10], SS (Average Staff) - Unit of measurement is Persons [9][10], while at the time, improved and increased P (Productivity) [9][10] of the software Rapid Application Development (RAD) process. So, decrease effort and time is key to success for any software development process. And ultimately, software Rapid Application Development (RAD) Model, and Software Project Management (SPM) is improved and focused.

5 Conclusion

Rapid Application Development (RAD) model is widely used in software industry but the number of activities being followed in different phases of RAD model is not of equal importance. Some of the activities are less important and thus contribute less in the overall output of RAD process. On the other hand, there are some activities which contribute more in the overall output of RAD process. These results are according to the experienced observations of the Software Practitioners implementing the Software Rapid Application Development (RAD) process model at work for the development of software applications and systems developed under organic mode of COCOMO model using the programming languages (C#, Java, Visual basic, and ASP). Our focus was to find less productive and more productive activities to give improved RAD model. We found that total activities are 115 in number. Out of these, 45 activities are less important/productive (39%), so there is no need to spend most of the time on these activities while 70 activities (61%) are critical/ productive, and must not be ignored as they are producing most of the output of RAD process.

This research resulted as the list of activities which produce most of the output using RAD model. So, software project manager can produce maximum outputs (high productivity) in less time/effort and other resources, by focusing on the improved Rapid Application Development (RAD) model shaped after the application of 80/20 rule. Whereas, the software Rapid Application Development (RAD) process model without the application of 80/20 Rule is looking like inefficient by the software professionals.

Our results show that just focusing on these 70 activities (our improved model) might be enough to focus at to get much (70 to 80%) of overall productivity and thus we can save the overall time and effort of software development, and performance or productivity will be increased. This will improve in result, the software RAD process, and software project management, because focus of the software practitioners will be maintained during whole software development life cycle at work using our improved software Rapid Application Development (RAD) Model in term of improved work-task part.

6 Case Studies and Future Work

The results of this research have been used in the development of software project using Rapid Application Development (RAD) Model for software(s) development in organic mode using languages (C#, Java, Visual Basic, and ASP). Software project managers have found improvement in the project management quality parameters while using our proposed activities to get more productivity at work. We have extended our research on the Rational Unified Process (RUP) Model, to be found in next publication. Our idea can be implemented in other fields like software estimation models and software risk mitigation by applying 80/20 rule to give improved models for the betterment and optimization of software engineering.

References

1. Ultsch, A.: Proof of Pareto's 80/20 Law and Precise Limits for ABC-Analysis, Data Bionics Research Group University of Marburg/Lahn, Germany (2002)
2. Koch, R.: The 80/20 Principle. Living the 80/20 Way. In: Nicholas Brealey Publication, Melbourne Australia (2004)
3. Futrell, R.T., Shafer, D.F., Shafer, L.I.: Rapid Application Development (RAD) Model Activities. In: Quality Software Project Management, 3rd edn., pp. 269–271. Pearson Education Publication, India (2004)
4. Pressman, R.S.: The RAD Model. In: Software Engineering – A Practitioner's Approach, 4th edn., pp. 34–37. McGraw- Hill Companies Publication, Inc., USA (1998)
5. Martin, J.: "Rapid Application Development". In: 'Rapid Application Development' (A Developers.net Publication) (1991)
6. Cooper, I.: Using MS EXCEL for Data Analysis and Simulation. Published in Science Teachers' Workshop, The University of Sydney (2002)
7. Varghese, C.: MS EXCEL: Statistical Procedures. Published in New Delhi, India (2007)
8. Kuhl, Project Lifecycle Model: How they differ and when to use them (2002)
9. Futrell, R.T., Shafer, D.F., Shafer, L.I.: Function Points Analysis. In: Quality Software Project Management, 3rd edn., pp. 325–337. Pearson Education Publication, India (2004)
10. Futrell, R., Shafer, D.F., Shafer, L.I.: Basic COCOMO. In: Quality Software Project Management, 3rd edn., pp. 376–379. Pearson Education Publication, India (2004)
11. David Consulting Group, QSM Function Point Programming Languages Table. A Publication of Quantitative Software Management, Inc. (2005)
12. Wright, S.: "Software Installation Plan, SIP (1999)
13. PRUTT, Software Lifecycle Models A publication of Adrian ALS & Charles Greenidge (2005)
14. Rizwan, M., Iqbal, M.: Application of 80/20 Rule in Software Waterfall Model. In: Proceedings of ICICT, 2009. IEEE Computer Society Press, Los Alamitos (2009)
15. Maner, W.: Rapid Application Development (1997),
<http://www.cs.bgsu.edu/maner/domains/RAD.htm>

Development of a Framework for Applying ASYCUDA System with N-Tier Application Architecture

Ahmad Pahlavan Tafti¹, Safoura Janosepah¹, Nasser Modiri²,
Abdolrahman Mohammadi Noudeh³, and Hadi Alizadeh⁴

¹ Computer Department, Majlesi Branch, Islamic Azad University, Isfahan, Iran

² Computer Department, Zanjan Branch, Islamic Azad University, Zanjan, Iran

³ Science and Research, Islamic Azad University, Tehran, Iran

⁴ Khayam University, Mashad, Iran

ahmad.pahlavantafti@poug.org

Abstract. N-Tier architecture considered as a comprehensive and integrated solution for designing, creating, developing and maintaining the large scale applications. Each application is a set of several physical and logical components. If we want an enterprise application, we should design these components as a well formed layout. N-Tier application architecture provides a model for developers to create a flexible and reusable application. By breaking up an application into tiers, developers only have to modify or add a specific layer, rather than have to rewrite the entire application over. There should be a presentation tier, a business or data access tier, and a data tier. [1]

Using N-Tier architecture would improve flexibility, reliability and extensibility software applications.

ASYCUDA¹ is the client/server application which performs customs declaration and clearance in some countries. ASYCUDA should assist Customs Administrations' modernization and reforms, by supporting both facilitation of legitimate trade and efficiency of Customs clearance controls [2]. It has implemented as a national project in Islamic Republic of IRAN Customs administration since 14 years ago. Nowadays, more than 90% of IRAN Customs processes are done with this system [11].

In this paper we analyze and adapt ASYCUDA application with the N-Tier architecture. We present the integration and validation of ASYCUDA application's tier with the N-Tier architecture and we propose a framework for applying it with the N-Tier application architecture. Then we evaluate some results about our proposed framework and finally we found that, it will make ASYCUDA more flexible and scalable.

Keywords: ASYCUDA, N-Tier Architecture, Customs Application, Scalability, Availability, Integrity.

¹The Automated System for Customs Data is a computerized system designed by the UNCTAD to administer a country's customs.[2].

1 Introduction

Software application is a product which design and create by the software engineers. The processes of design and create a software application are depend on the architecture which they used. Wherever we face to high complexity and large size application, it is necessary to use special architecture. All of the software architectures include the structure of components, relations between components and principles guidance which manage the design and develop of a system [3].

N-Tier architecture is a contemporary model in software engineering. Whenever we use N-Tier model, the modification of the application is easier than other monolithic tier. In N-Tier architecture, the entire core of an application divides into some parts. When we break up an application into tiers, programmers and developers could alter or add a specific layer, rather than the entire application. An application is a set of some parts. Each application must provide ability to store and access data in a consistent manner with a well formed model and it must use a standard graphical user interface and functional processes. A tier is a reusable part of code in an application which performs a specific task. With N-Tier architecture we achieve a flexible and scalable application in a cost effective way [4]. These advantages of this model are caused to choose N-Tier architecture as a goal of this article, indeed.

ASYCUDA is an automated system for customs data and it is a successful application in the filed of the optimizing customs duties.

In this paper, we investigate on the corresponding of ASYCUDA system and N-Tier architecture. The reminder of this paper arranged as follows. The principles of the N-Tier architecture are described in the next section. Comparing ASYCUDA with three types of N-Tier architectures is described in section 3. Proposed framework for applying ASYCUDA with N-Tier architecture is shown in section 4 and evaluation of this framework is described in 5th section. Section 6 illustrates the results and conclusion and the future work are addressed in section 7.

2 N-Tier Architecture

In software engineering, N-Tier architecture is a client-server model in which the presentation, the application processing and the data management are separate logical processes. The most widespread types of N-Tier architecture are the three, four and five-tier architecture.

2.1 Three-Tier Architecture

It was developed by John J. Donovan in Open Environment Corporation (OEC) and it is intended to allow any of the three tiers to be upgraded or replaced independently as requirements or technology change [1]. It has the following tiers:

- **Presentation Tier**, This is the top level of the application and interacts with the users. It means that the end users only work with this tier. It communicates with other tiers via the SOAP, XML and RPC protocols and technologies [5].

- **Business Tier**, It is the logic core of an application. It controls an application's functionalities and responsibilities. All of the application's processes surrounded with this tier. EJB, .NET, CORBA, COM+ and DCOM are the famous technologies and platforms which are used in business tier [6].
- **Data Tier**, This tier performs data storage and retrieval. It is a data repository like the databases or the file systems [5]. Fig.1 shows three-tier architecture.

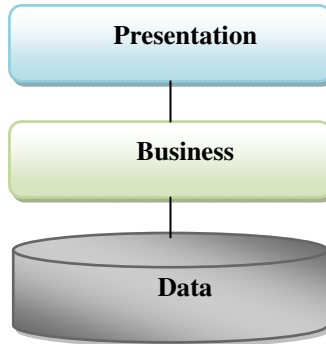


Fig. 1. Three-Tier Architecture

2.2 Four-Tier Architecture

Four-tier architecture is based on the three-tier architecture and pushes one tier between the business and data tier, as a data access tier. The data access tier is responsible for managing and controlling the data tier [7]. You can see this architecture in Fig.2.

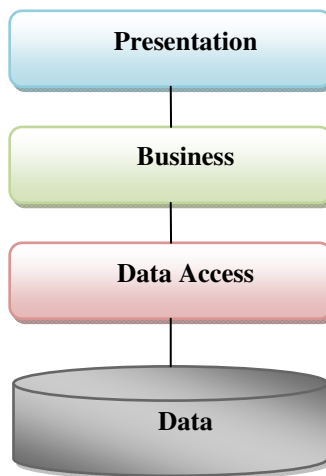


Fig. 2. Four-Tier Architecture

2.3 Five-Tier Architecture

As you can see in Fig.3, this architecture is as same as the four-tier architecture, but it adds a tier on the topmost level. This tier is visualization tier and formats the layouts of an application [8]. Personal computers are the common interface which users work with them. Nowadays, we encounter with other devices like mobile, PDA and so to interact with an application and we accept the differences between these devices and personal computers. Visualization tier is suitable for this purpose. It is physically established on the client machine.

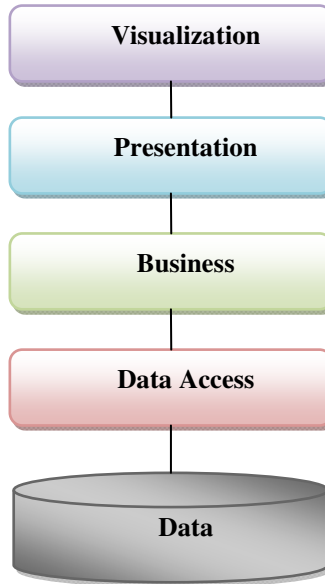


Fig. 3. Five-Tier Architecture

It is better to disturb each tier separately in N-Tier architecture. This architecture has invaluable properties like scalability and flexibility. Distributing each tier on a separate machine could increase the performance and decrease the response time, especially when we have many concurrent users. N-Tier architecture is based on some rules. Each tier could communicate with only its top or bottom tier and could replace with the different tools and technologies.

3 Applying ASYCUDA with N-Tier Architecture

In this section we investigate on applying ASYCUDA system with N-Tier architecture. As we mentioned earlier, ASYCUDA is an application that performs main customs functionalities. ASYCUDA is installed at the request of developing country governments with the assistance of UNCTAD experts [2].

ASYCUDA system migrated customs offices to a modernized customs. ASYCUDA has participated for many years in various working groups and organizations for normalization and it is particularly involved with the WCO working group on the Custom Data Model that has now become simply Data Model after its opening to Other Governmental Agencies (OGAs) [2].

The main objective of the WCO Data Model is to define a set of standardized Data usable by both Customs Administration and trade operators, for electronic data exchange within the Customs Clearance process (manifest, declarations ...) [2].

Is ASYCUDA corresponded with N-Tier architecture? In Table 1, you can see the true and the false alerts of corresponding between the ASYCUDA system and N-Tier architecture. As you see, ASYCUDA is not applied with five-tier architecture.

Table 1. The true and the false alerts for applying ASYCUDA with N-Tier Architecture

Applying ASYCUDA with N-Tier Architecture					
Architectures	Tiers				
Three-Tier	Data	Business		Presentation	
	TRUE	TRUE		TRUE	
Four-Tier	Data	Data Access	Business	Presentation	
	TRUE	TRUE	TRUE	TRUE	
Five-Tier	Data	Data Access	Business	Presentation	Visualization
	TRUE	TRUE	TRUE	TRUE	FALSE

4 Proposed Framework

Table 1 shows the compatibility between the N-Tier architecture and ASYCUDA system. According to the false alert in visualization tier, between ASYCUDA and five-Tier architecture, we attempt to find architecture to improve quality of the proposed framework. In this framework, we add an abstract tier for visualization the user interface. All of these tiers are separated in the framework. Fig.4 illustrates the model of the framework.

Visualization tier in our proposed framework aims to reformat the user's request and response and permits the users could communicate with ASYCUDA with any computer devices like PDA, Personal Computers, Laptops and etc. the WAP and IMAP4 usually reformat the user interface for PDA and Mobiles computers. XSLT and XPATH are a language that describes how to locate specific elements (and attributes, processing instructions, etc.) in a user documents. It allows users to locate specific content within an XML document [9].

Recommendation framework attempts to show a set of modern technologies with the highest degree of abstraction and connectivity between tiers.

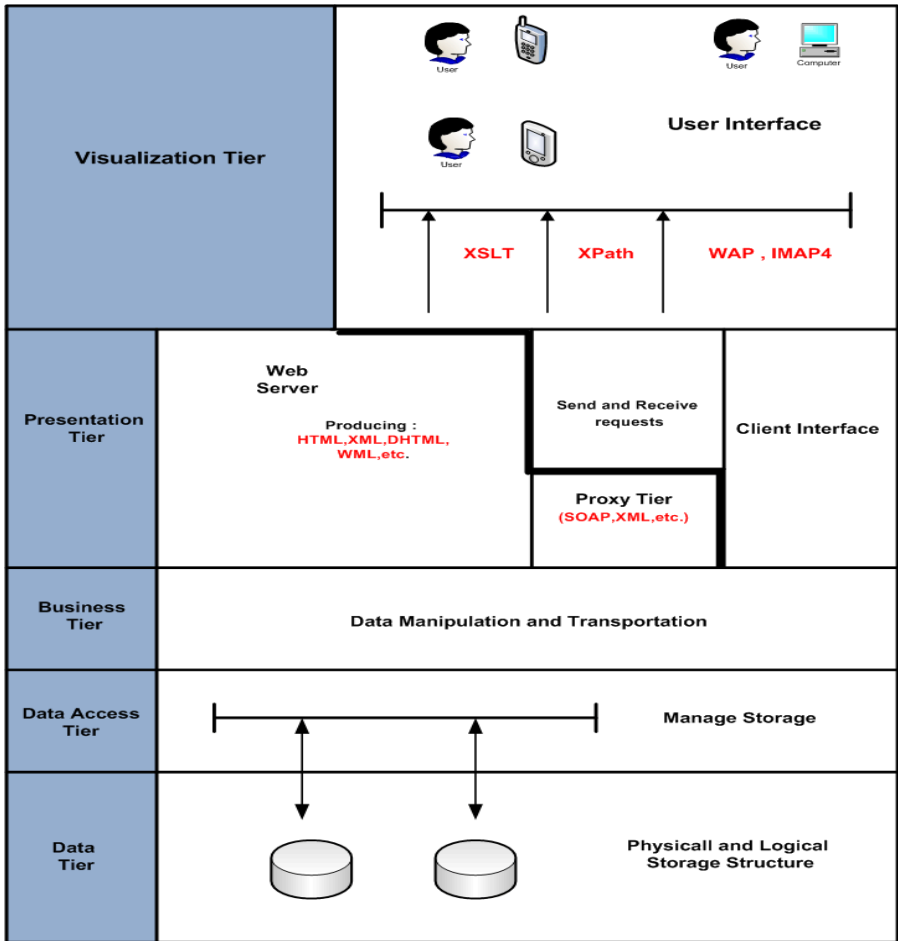


Fig. 4. Proposed Five-Tier Framework for ASYCUDA

Proxy is an object or person authorized to act for another [10]. In our framework, Proxy is referring to any sort of code that is performing the actions for clients. The client interface is connected directly to the business tier.

The topmost level of this architecture incorporates the user interface to present output information and also to route the input parameters as well as user actions to the presentation Tier.

5 Proposed Framework Evaluation

Different types of software properties are used to evaluate the capabilities of the recommendation framework. We listed several software architectural properties in order to evaluate the characteristics of our proposed framework. The software architectural priorities which we have classified are Flexibility, Openness, Modularity

and Scalability. We choose 50 persons of internal ASYCUDA users, 50 persons of external ASYCUDA users and 50 persons of the ASYCUDA domain experts and 50 persons of software architecture domain experts. We choose Khorasan (Mashhad) customs Administration office for our dataset. It is a branch of Islamic republic Of Iran Customs Administration (IRICA).

Opinions of the Internal and external users of ASYCUDA system are shown in Figure 5.

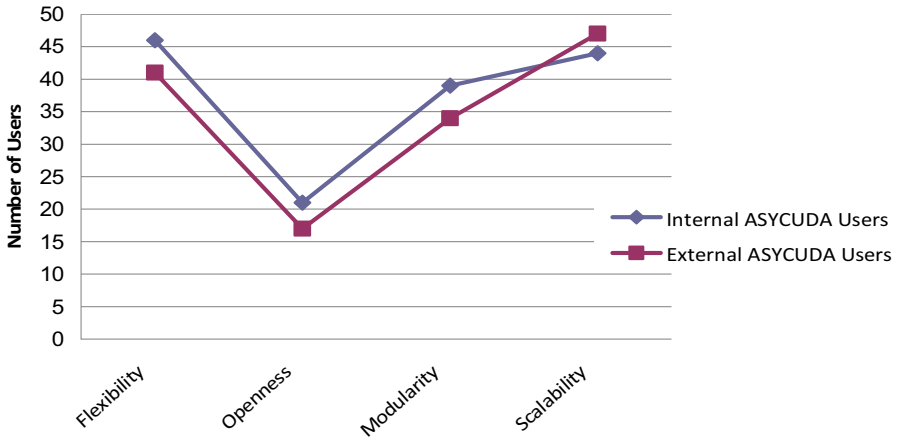


Fig. 5. Internal and External Customs User's Viewpoints

Figure 6 shows the evaluation of the domain experts.

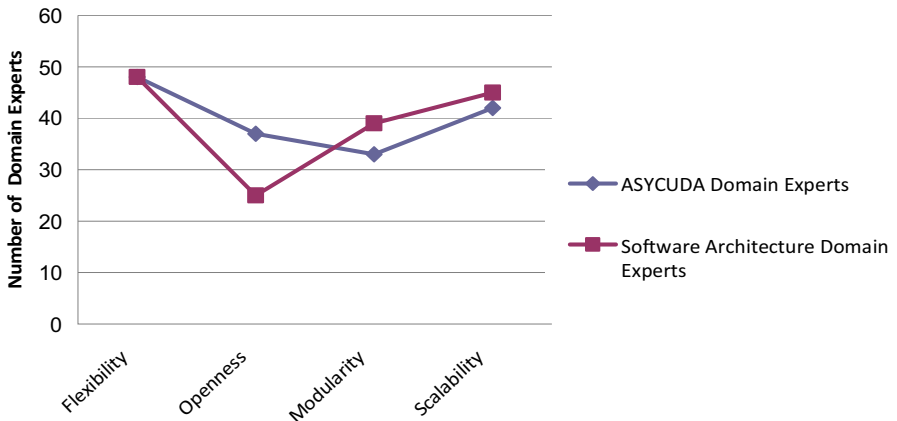


Fig. 6. Evaluation of the domain experts

6 Results

Figure 5 and Figure 6 demonstrate the results of the polls which have done at Khorasan Customs. As you see, these diagrams display some fluctuations percentage of some properties that implemented by the proposed framework.

Figure 5 illustrates that our framework is more scalable in external users' opinions, about 95%, while it is approximately 80% for internal users. Also this diagram demonstrates some parameters such as openness, modularity and scalability which the external users believe that it is more scalable, approximately 95% whereas it is about 88% for internal users. The percentage of openness and modularity is lower than the others. For instance, results of the evaluation show that the openness of our framework is less. Also percentage of modularity in proposed framework is about 78% for internal users and 70% for external users.

Figure 6 displays variety of many experts' viewpoints about percentage of some architecture's characteristics. According to our evaluation, a large number of experts almost 98% believe that the system is flexible and more than 85% of them believe that our proposed framework is scalable, but the system is not openness enough almost 50%. The people who have some experiences in ASYCUDA system and customs domain believe that the system has less modularity, approximately 68%.

Our proposed framework makes more flexibility for ASYCUDA application in a scalable manner.

7 Conclusion and Future Work

The software applications of customs must be as comprehensive, open and integrated tools which are able to address the needs of expected business.

To design and create a more efficient customs application that will facilitate trade, we represent an integrated solution for applying ASYCUDA system with a flexible and scalable architecture.

Multi-Tier architecture has several benefits such as integrity, availability, flexibility and scalability. In N-Tier model, every layer is independent from others; it means we could modify each layer easily, without changing any thing in other layers.

In this paper, we used the advantages of N-Tier architecture and improve ASYCUDA architecture as a customs digital system. We design a framework as Five-Tier architecture to increase the ASYCUDA capabilities in scalable and flexible manners.

In the future we will study about new generation of customs applications and we will represent a software architecture which is suitable for modern customs.

References

- [1] <http://en.wikipedia.org>
- [2] <http://www.asycuda.org>
- [3] Edwards, J., Devoe, D.: 3-Tier Client/Server at Work. John Wiley & Sons, Inc., Chichester (1997)
- [4] Tanenbaum, A., Steen, M.V.: Distributed Systems, 2nd edn., Prentice-Hall, Englewood Cliffs (2007)
- [5] http://en.wikipedia.7val.com/wiki/multitier_architecture

- [6] <http://publib.boulder.ibm.com/infocenter/wasinfo/v5r0/topic/com.ibm.websphere.exp.doc/info/>
- [7] Mckinely, P., Sadjadi, S., Kasten, E., Cheng, B.: Composing Adaptive Software, January 2004. IEEE Computer Society Press, Los Alamitos (2004)
- [8] http://en.wikipedia.org/wiki/visualization_computer_graphics
- [9] Nanda, M., Karnik, N.: Synchronization Analysis for Decentralizing Composite Web Services. International Journal of Cooperative Information Systems 13(1) (2004)
- [10] <http://www.m-w.com/cgi-bin/dictionary?proxy>
- [11] <http://www.irica.gov.ir>

An Experimental Design to Exercise Negotiation in Requirements Engineering

Sabrina Ahmad and Noor Azilah Muda

Faculty of Information and Communication Technology,
Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya,
76100 Durian Tunggal, Melaka, Malaysia
{sabrinaahmad, azilah}@utem.edu.my

Abstract. A framework design is critical to the deployment of empirical investigation. Safety measures must be taken in order to ensure that the data obtained from the study is reliable and the measures taken are valid. This paper provides a framework design to deploy an empirical investigation to assess the effectiveness of negotiation in requirements engineering. It also elaborates the relevance of negotiation in requirements engineering process and its effectiveness. The underlying concept and the motivation which influenced the framework design are stated. The framework design which is divided into population and participants, the experimental process, and methods to minimize bias is thoroughly explained.

Keywords: Empirical study, software requirements, requirements elicitation, negotiation.

1 Introduction

Software requirements quality is usually assessed through verification and validation of intermediate or final product. The requirements are checked against requirements specification, prototypes or the end product. This is known as an analytical approach. It describes an effort to detect the defects within the software development products and fix them.

Meanwhile, a constructive approach is applied while developing the requirements. This approach suggests prevention to ensure that mistakes are minimized during the creation of requirements. In this research, a constructive approach was adopted by enforcing negotiation in the requirements elicitation process. Negotiation is seen as a preventive action whereby defects are not introduced into the requirements statement. Therefore, the requirements elicitation process which incorporates negotiation is expected to list better quality requirements.

The rationale of adopting the constructive approach and measuring the quality at a very early stage is obvious. History tells us that the greatest number of errors and the errors that are most costly to fix are generated at the beginning stage of software development process. Errors in requirements are the most numerous in the software lifecycle and also the most expensive and time-consuming to correct [1]. The context

in which requirements are elicited is usually a human activity, and the problem owners are people. It is seldom technical problems which inhibit productivity and quality [2, 3]. Instead the vast majority of requirements problems are related to human interactions, process and communications. One of the main problems during requirements elicitation is communication and understanding among the stakeholders. This involves conflicts, scope boundary and erroneous interpretation. The argument is supported by Zowghi [3] who believed that requirements elicitation is inherently imprecise as a result of multiple variable factors, a vast array of options and decisions, and communication.

Due to the urgency of quality requirements for quality software, this paper outlines an experimental design to allow empirical investigation in order to implement negotiation in requirements engineering. The design of the experiment also explains the mechanism of negotiation activities which contribute to the improvement in requirements quality.

2 Negotiation in Requirements Engineering

In a process of identifying the right requirements to develop, conflicts are common since stakeholders frequently pursue mismatching goals. Reaching agreements among stakeholders who have different concerns, responsibilities, and priorities is quite challenging. Therefore, negotiation is useful to handle the conflicts and to resolve disagreement between the stakeholders. According to Grunbacher[4], negotiation leads to benefits such as understanding project constraints, adapting to change, fostering team learning, revealing tacit knowledge, managing complexity, dealing with uncertainty and finding better solutions. Furthermore, the benefits of negotiation are obvious and many researchers have pointed out its usefulness for requirements engineering [5-11]. None of these studies measured the improvement in requirements after negotiation.

Based on literature, advantages for deploying negotiation are best classified in four categories. They are conflict handling, shared vision, cooperation, and knowledge. Negotiation contributes to **conflict handling** because it facilitates conflict detection and resolution [12]. Before an actual conduct of negotiation, requirements statements are examined to identify conflicts by analysing stakeholders' goals and preferences. The EasyWinWin negotiation approach identifies conflicts manually and relies on the knowledge and expertise of the involved stakeholders and the capabilities of the facilitator [13]. Other researchers have tried to automate or partially automate the task of understanding requirements conflict. For example, Egyed and Grunbacher[14] presented an approach for identifying conflict and cooperation among requirements based on software attributes and automated traceability. Another example is from Kaiya[15] who introduced a systematic approach to identify conflict through preference metrics in AGORA (attributed goal-oriented analysis). Sequentially, the identified conflicts are then negotiated to seek mutually beneficial solutions that are acceptable by the stakeholders. The negotiation contribution towards conflict handling is also proven empirically by several researchers through experiments [6, 9, 16]. Suppressing or overlooking conflicts is risky and might have serious negative effects on the software development process. Understanding requirements conflicts is

thus an important strategy to mitigate software development risks. This is supported by much literature emphasizing the importance identifying and analysing conflicts for the success of system development [17-21].

Negotiation also promotes **shared vision** among multiple stakeholders. The negotiation process addresses the stakeholders' concerns and thus establishes shared vision to achieve mutual understanding. This is supported by other researchers as they also claimed that one of the negotiation benefits is to establish shared vision [9, 22, 23]. Throughout a negotiation process, stakeholders share their interests of the requirements they need and thus provide understanding to other stakeholders. This process allows various stakeholders to acknowledge others' concerns for the benefits of the system to be developed. Usually, stakeholders contribute incomplete, vague, and often inconsistent statements and ideas about their objectives, assumptions, and expectations. As they work together to negotiate their requirements, they give the project shape, and their merged visions emerge into a system that other stakeholders can accept. If, on the other hand, the stakeholders do not negotiate together, there is little chance the resulting system will accommodate their needs and the project will often fail. Negotiation is, therefore, essential to achieve mutually satisfactory agreements.

The shared vision and the satisfactory agreement increase the level of **cooperation** and trust among the stakeholders. As negotiation processes explore the stakeholders concerns, needs and visions and their ideas towards developing a reliable and workable system are acknowledged. The acknowledgement leads to cooperation as the agreement is a group decision which recognize the various stakeholders viewpoints [24]. This is also proven by empirical study and reported experience in literature [6, 16, 25]. The cooperation among the stakeholders is important to support the development process along the way and to ensure the success of the system being developed. At the end of the day the developed system provides functions the stakeholders need to assist their business process.

Further, the negotiation process improves the shared **knowledge** gained by the stakeholders. Usually, stakeholders state their needs towards the intended system with an implicit knowledge of their own work. A statement can be easily misinterpreted or misunderstood by the others. Through the negotiation process, stakeholders need to explain and elaborate their requirements in order to provide understanding to others. In addition negotiation invokes the exploration of solutions before reaching agreement. Also, through negotiation, stakeholders are forced to justify the need of the requirements they request and the rationale of having the said requirement. The negotiation process therefore narrows the knowledge gap and reveals the tacit knowledge of the multiple stakeholders [5, 9, 26, 27].

3 The Proposed Framework of Negotiation Process

This section explains the essence of the framework design implemented in this research. Explains here are the high-level model of negotiation process, concepts and terms used throughout the research and related works which motivates the experimental design.

3.1 High-Level Model of Negotiation Process

A high level model is introduced here to give an overview of the interactions that happen to produce agreed requirements through negotiation. As illustrated in Figure 1, the input of the model is the candidate requirements which may or may not contain conflicts. Next, the process of conflict identification takes place. This is followed by conflict resolution in which the stakeholders define and share glossary, share perspectives, views and expectations on requirements, justify the need of the requirements, prioritize the requirements and assess the system feasibility. Out of these efforts, a group decision is achieved to produce agreed requirements.

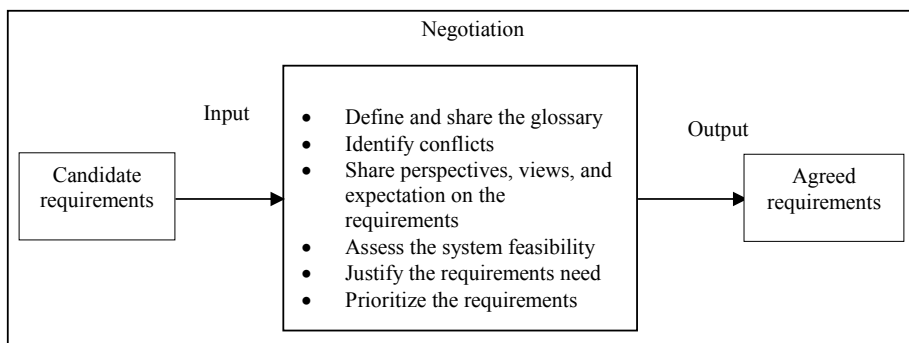


Fig. 1. The Negotiation Process Expressed as a High Level Model

Further details of the components of negotiation are as follows:

- Define and share the glossary* – This process allows the stakeholders to define and to share the meaning of important keywords. A clear and explicit definition yields the same interpretation used in the requirements statements. The same interpretation is useful to assist multiple stakeholders to understand definitions without ambiguity. There are at least two literature which support the fact that sharing the glossary is important to prevent inconsistency in interpretation [13, 15]. EasyWinWin methodology comprises activities of gathering, elaborating, prioritizing and negotiating requirements. Additionally, in order to avoid the occurrence of misinterpretation, EasyWinWin includes the ‘capture a glossary of term’ sub-activity wherein stakeholders can define and share the meaning of important terms and words appearing in the requirements statements. AGORA adopts a scoring technique that initially focuses on vertical conflicts in preference matrices in order to systematically find discordances in interpretations.
- Identify conflicts* – Negotiation process focuses on conflicts identification to gather the attention of the stakeholders on problematic requirements. This effort motivates them to work together in order to find a resolution. Conflicts do not necessarily contain defects but may contain possible defects which are worth unfolding, justifying and assessing thoroughly. There are at least three literature which agree and prove that conflict identification is useful to identify possible

defects, which in turn leads to resolution. Boehm [23] who introduced EasyWinWin as a tool based on negotiation methodology incorporates an activity called '*Identify Issues, Options, and Agreements*' to register conflicts as a foundation from which to propose resolution options and therefore provides the foundation to negotiate agreements. Kaiya[28], who introduced AGORA provides systematic conflict identification through preference matrices value, also proved that conflict identification is useful to identify which requirement should be improved and refined. Robinson et al [29] introduced conflict-oriented approach to identify and to remove conflicts in order to have a better structure requirements.

- *Share perspectives, views, and expectations of the requirements* – This process allows the stakeholders to clarify and to further elaborate the requirements statements. The clarification and further elaboration of the requirements revealed the stakeholders perspectives, views and expectations towards the requirements statements. Grunbacher [9] stated that people know more than they can tell. Also, implicit stakeholders' goals, hidden assumptions, unshared expectations often result in severe problems in the later stages of software development. There was at least one literature on negotiation method which supported the fact that sharing perspectives, views, and expectations of the requirements was important to reveal tacit knowledge. EasyWinWin includes the '*Brainstorm stakeholder interests*' to allow the stakeholders to share their goals, perspectives, views, and expectations by gathering statements about their win conditions. This activity had proven beneficial during implementation using real-world negotiation as reported in [9].
- *Assess the system feasibility* – This process allows the stakeholders to assess the system feasibility from the perspective of resource feasibility and dependency feasibility. Resource feasibility means that the requirements are assessed if the subset of requirements can be built within time and cost constraints. While dependency feasibility means that all requirements in the subset are assessed if all the dependencies are included in the subset. This effort assists the stakeholders to make informed decision on the practicality of the agreed set of requirements. There were at least two literatures on project management which supported the fact that assessing system feasibility was important to ensure the success of the software project [30, 31]. The literature discussed the scenario of the possibility of an infeasible system if the project resources were not considered during requirements engineering process.
- *Justify the requirements needs* – This process allows the stakeholders to justify the needs and the importance of the requested requirements. The stakeholders need to think through the requirements and consider why one requirement is more important than the other in order to justify them to other stakeholders. There was at least one empirical evidence which supported the fact that requirements justification forced the stakeholders to think thoroughly on the requirements need and importance. In a small team, negotiation is exercised and based on observation; it is reported [32] that the negotiation process forced the participants to justify the need on every request demanded in order to gain other participants' understanding.
- *Prioritize the requirements* – This process allows the stakeholders to prioritize the requirements statements, to define and narrow down the scope of work and to

gain focus. Prioritization makes it possible to gauge the importance a client feels regarding each requirement in respect of a software solution being able to fulfil their needs. There was at least one literature on negotiation method which supported the fact that requirements prioritization managed to narrow down the focus which assisted in group agreement [13]. Through prioritization [33], if it is not feasible to complete all of projects requirements, it is still possible to see which requirements are most important to the customer and implement those before the less important ones. This means that a project which has not had all its requirements fulfilled can still be of high value to a customer when it has fulfilled the customers' most important requirements. In an example, Karlsson et al [34] showed that 94% of the project value can be delivered for about 78% of the possible maximum cost.

3.2 The Underlying Concept

Mohammed [35] stated that having agreement between parties is paramount. Negotiation is deployed in this research to achieve agreement between the system's stakeholders in order to identify a set of requirements to be developed. Negotiation is usually understood to be a bargaining process between two or more parties to identify or to resolve people's needs of a system. A common bargaining process is between customer and developer to agree on the requirements to be developed and the project cost and time. The objective is to achieve an agreement on a business deal and then to proceed with the agreed software development.

Four key concepts need to be emphasized here as these are the concepts applied in the research:

Consensus-based negotiation is applied in this research in which the system's stakeholders, working together, reach group objectives rather than compete against each other. The group objective is mainly the development of a system which benefits the organization and at the same time represents the key stakeholders' perspectives and perceptions [27]. The main concern with regards consensus is not to reach unanimity but rather that all the stakeholders are committed to accept the consensual decision and feel that their perspectives and ideas are acknowledged in a cooperative manner. The consensus decision making is adopted because it is based on the belief that each stakeholder has some part of the truth while no one person contributes all. It is also based on a respect that all persons involved in the decision making be considered. Consensus enables a group to take advantage of all group members' ideas. It is a reasonable expectation that a decision based on a combination of thoughts would be of a higher quality than any individual decision. Choudhury et al [36] stated that working in a group provides a wide range of advantages by sharing information, generating ideas, making decisions and reviewing the effects of the decisions. Ideally, the group will reach a better decision than an individual because collective knowledge and expertise of the group is greater than that of any individual. Further, people are more likely to implement and accept decisions they have accepted by consensus [36, 37].

Consensus-based negotiation may be summarized as:

- Agreement on the decisions by *all* the key stakeholders;
- Acceptance of consensual decision and acknowledgement of the stakeholders' perspectives and ideas;
- Respect for all persons involved in the decision making process;
- Use of the collective knowledge and skills of the group and
- Creation of collaborative environment among the stakeholders.

The **stakeholder** is a term that refers to any person or group who will be affected by the system directly or indirectly. Stakeholders include end users who interact with the system and everyone else in an organization that may be affected by its installation. Other system stakeholders may be engineers who are developing or maintaining related systems, business managers, domain experts and trade union representatives [38]. However, it is inappropriate and impossible to have all of the system stakeholders in the requirements elicitation process. It is impractical to involve a huge number of people in a face-to-face negotiation process. Negotiations practice [6] usually involves the key stakeholders (also known as success-critical stakeholders) to determine success. These stakeholders are the key people to represents their group interests and may include the end users, the system owner and managers who collaborate and are actively involved in decision making to achieve mutually satisfactory agreements. Therefore during the empirical study, only the key stakeholders involve to represent the key people.

The '**silent objective**' is enforced to the empirical study. It means the researcher's purpose for the experiments will be not revealed to the participants performing the negotiation. The 'silent objective' is not revealed in the experiments' instruction to the participants. This 'silent objective' is employed to allow the participants to merely exercise negotiation without knowing the underlying objective of the researcher. If the objective is revealed, the participants will tend to prioritise wrongly by striving to achieve the objective without having negotiation. This is to ensure that this research is purely assessing the negotiation process and the results obtained from that process.

A **defect** is defined by the researcher as summarized here. The literature is rife with inconsistent usage of this term. For example, McConnell [39] makes no distinction between errors and defects in the examples he cites in his book. On the other hand, Humphrey [40] elaborately states a bug is a defect but not all defects are bugs, and all defects result from errors but not all errors produce defects. Even the software measurements collected by authoritative organizations reflect a lack of consensus; Christensen et al [41] stated that NASA and DoD used the term "defects" while the Software Engineering Laboratory refers to "errors" and the Army refers to "faults" and "anomalies". Pressman [42] define defect as a deviation between the specification and the implementation, detected after release to the customer (or the next activity in the software process). This is supported by a definition in IEEE [43] and SWEBOK [44] in which the standard define defect as product anomaly and a quality problem discovered after the software has been released to end-users respectively. These definitions fit the big picture of software development in which the specification can be checked against the end product to recognize the existence of defect or not.

This research is using the term defect to represent requirements defect which may occur during requirements elicitation process. However, defect in this research refers to the nonconformance of requirements at a very high level of requirements elicitation phase. As this research has a limitation within RE phase only, the general definition of defect as stated above is not appropriate. The requirement defect at this stage is checked within the written requirements statement and against the conformance of the stakeholders needs. Aligned with that, at this stage, only a number of defect attributes which associate with several quality attributes is relevant. Lauesen et al [45] looked into the effort to prevent defects early in the process life-cycle, defined requirement defect as “although the product works as intended by the developers, the users and customers are not satisfied with it. They may find it too difficult to use or unable to support certain user tasks. Unstated user expectations (tacit requirements) and misunderstood requirements are typical examples”. Similar research [46-48] which looks into requirements defects in this early stage line up more or less the same defect attributes in their research. Therefore, by definition, a defect is a nonconformance of requirements in requirements statements and customers’ needs based on the requirements comprehensibility, completeness, consistency, feasibility and correctness. Customers’ needs are represented by the high level requirements statements listed as agreed requirements following negotiation.

3.3 Related Works

This sub-section elaborates the motivation which influenced the negotiation process introduced in this research. The process was designed to provide negotiation facility during the requirements elicitation process among multiple stakeholders. The basic features were conflict detection and resolution, requirements exploration and requirements prioritization to assist in achieving group decision. Discussed below are current methods and techniques which motivate the negotiation process introduced in this research.

In terms of conflicts and misinterpretation detection, EasyWinWin[23] is identified as a useful negotiation methodology with collaborative tools which provides electronic brainstorming, categorizing and polling. It includes the “capture a glossary of terms” sub-activity wherein stakeholders can define and share the meaning of important terms and words appearing in the requirements statements. This effort requires the stakeholders to create a record of the glossary. Once the glossary is recorded, it can be viewed by all the stakeholders involved in the negotiation. Also, EasyWinWin has a tool called quality attribute risk and conflict consultant (QARCC) which systematically provides suggestions to the stakeholders regarding the possibilities of potential conflicts by using a knowledge base. In the knowledge base, pairs of conflicting quality attributes are stored. However, the success of this approach largely depends on the quality of the knowledge base and in general it is a huge effort to build such a knowledge base. The development of the knowledge base needs project backgrounds, documentations and histories of previous projects to allow archiving and mapping on the potential conflicts. This effort is useful if only the organization has the history of previous projects. What if a knowledge base is not available? The approach introduced to detect conflicts in this research does not require such a knowledge base in advance. This research adopts a scoring technique

to systematically detect the conflicts. In this activity, individual stakeholder need to assign a score value for every requirement based on individual preference. Whenever the scores differ, there are conflicts. Even though the approach used in this research does not require knowledge base as in EasyWinWin, the benefit of knowledge sharing among the stakeholders emphasized in EasyWinWin is noted.

Attributed Goal-oriented Analysis (AGORA)[15] introduced a scoring technique that focused on vertical conflicts and diagonal conflicts in preference matrices. Vertical (off diagonal) conflicts systematically find conflicts in interpretations and diagonal (the main diagonal of the matrices) conflicts systematically find conflicts in stakeholders' interest. However, AGORA requires a well trained facilitator to facilitate the requirements elicitation process who understands how AGORA works and who is capable of handling the entire process. Also, during the process with AGORA, the stakeholders need to guess what other stakeholders think of every requirement and assign a score to it. If the variance of the score is high, it is believed that there might be conflicts in interpretation with the requirement and further elaboration is required. On the other hand, the approach in this research does not require a trained facilitator to assist the elicitation process because neither tools nor complicated graphs nor matrices are used.

This research adapts and simplifies the scoring technique in AGORA [28] to detect the conflicts in preference among multiple stakeholders. The vertical conflicts which identify interpretation issues are not included as misinterpretation and inconsistent conflicts are managed in the face-to-face negotiation process which reveal tacit information and shared common understanding.

The scale of scoring technique used in this research is adapted from the MoSCoW technique [49]. MoSCoW is a prioritisation technique used in business analysis and software development to reach a common understanding with stakeholders on the importance they place on the delivery of each requirement. The capital letters in MoSCoW stand for:

M - MUST have this.

S - SHOULD have this if at all possible.

C - COULD have this if it does not affect anything else.

W - WON'T have at this time but WOULD like in the future.

Table 1. The Scale for Requirements Prioritization

Scale	Meaning
4	Must have this
3	Should have this if at all possible
2	Could have this if it does not affect anything else
1	Will not have this time but would like in the future
0	Must never have this

In this research, this method was converted into a numbered scale from 0 to 4 in which an item was added to scale 0 meaning 'Must never have this.' This item was introduced to provide an option if the stakeholders do not want the particular requirement to be included. This is possible in a circumstance of requirements which

are requested by a stakeholder but is not wanted by the other. For example, lecturers would like to have their students' photos to be tagged along the electronic report card for prompt recognition but on the other hand the students are not comfortable to have their photos online. In this example, the lecturers' representative suggests a requirement to have the students' photos online but the students' representative choose to exclude the requirements. Hence, the 'Must never have this' is the best option to represent the students' preference. Table 1 below state the scale used in this research.

A cycle of explanation and elaboration in the negotiation phase in this research is designed to promote understanding, to allow the stakeholders to make informed decisions and therefore achieve consensus. This approach is influenced by Delphi technique which is usually used to survey and to collect the opinions of experts. The Delphi technique is widely used and accepted method for gathering data from respondents within their domain of expertise. The technique is designed as a group communication process which aims to achieve a convergence of opinion on a specific real-world issue [50, 51]. The strength of Delphi, in contrast to other data gathering and analysis techniques, employs multiple iterations designed to develop a consensus of opinion concerning a specific topic via questionnaires. It is noted that Delphi usually keeps the stakeholders isolated. However, this research adapted the iterative process of Delphi to converge the stakeholders' opinions in face-to-face iteration format. This activity allows information sharing emphasizing the justification of the "need" or "not need" of the software requirements.

3.4 The Framework Validation

The framework will be validated through role play experiments exercising negotiation in a software development project simulation. A pilot trial is carried out to check and improve the experimental design before an actual experiment takes place. The trial includes a post-mortem session to collect the participants' feedback. Experimental noise is revealed by this feedback (if any) and necessary improvements will be made. Then, improved experimental design will be used to deploy actual experiments. The data from the experiments will be measured to validate the framework.

4 The Experimental Design

This section describes the framework design deployed in the research based on guidelines by Kitchenham et al [52]. It provides guideline and control on the population being studied, the rationale for sampling from that population, the process for allocating and administering the experiments, and the methods used to reduce bias. Throughout this section, trials and experiments are mentioned several times but the experiments are not reported in this paper as it focuses on the framework.

4.1 Experimental Subjects

This sub-section defines the population from which the participants for the experiments were drawn, the process by which the participants were selected and the process by which the participants were assigned to the experiments.

The experiments were done in a series of tutorial sessions at The University of Western Australia. Two course units with at least 20 people each were involved in two semesters to allow several trial run and the actual experiments to take place. The units were Software Requirements and Project Management (CITS3220) and Software Engineering Industry Project Leadership (CITS4222). The units shaped the students to become effective team members, undertake problem identification, formulation and solution and apply their knowledge of basic science and engineering skills

These two units were identified as the most suitable units to provide the right group of students with the right level of knowledge to deploy the experiments for this research. These were students with a software engineering knowledge background. Particularly they were equipped with the theory and concept of negotiation through formal lecture before the experiment. Some had working experience in software development.

In order to avoid the presence of bias, the participants' assignment to the experimental groups and to the role they were playing in the actual experiments was random. The participants who had special ability, such as people with working experience or a high achiever, were identified by the unit coordinator and divided evenly among groups. This was done to avoid the possibility of having a distinguish group which consist of brilliant participants who would produce very good negotiation results. Good results may not represent the effectiveness of negotiation but simply the participants' intelligent guesses. Hence, in this research, on top of random group assignment, extra effort to avoid the presence of bias is necessary.

In addition, a role play experiment always comes with the dilemma of whether the participants are really playing a role or simply incorporating their personal judgment. Expecting that each participant would be more committed to a specific priority when given a clear role and in order to minimize that possibility, the participants were given instruction and guideline on how to play the role of the system's stakeholder. In addition, to assist the participants to feel the responsibility of being the system's stakeholders, the description scenario and the candidate requirements were given to them in advance. These reading materials helped because the description scenario described the need of the system and the concern of different stakeholders and the candidate requirements were carefully tailored to the specific stakeholder's needs. In addition, observation done by the researcher, her supervisor and unit coordinator throughout the experiment discovered that most of the participants were playing the role given to them; this is due to the peer assessment for the unit of the tutorial session where the experiments were done.

4.2 Experiment Procedures

This sub-section defines the experimental unit, describes the study design and explains the experimental process.

Each experimental unit was a group of four or five participants exercising negotiation. The number of groups available for each experiment was treated as a replication of the treatment. Every experiment involved four to six groups exercising negotiation. Hence, negotiation was essential in all experiments and exercised by all

the groups. The results from the experiments produced a list of software requirements which had been negotiated among the participants within a group and measured respectively.

Initially in the experimental process, all handouts such as the instruction sheet, the description scenario, the candidate requirements and the decision forms, were given to the participants. Next, a briefing on the background knowledge of the experiment took place. This was followed by instructions which were supported by a sample overview of step-by-step activities. The participants' assignment to the groups with the role to play during the experiment was then given and followed. Ample time was given to the participants to understand the roles and the candidate requirements prepared for them. The participants were then asked to make an individual decision based on resource constraints on which requirements should be implemented. This activity acted as a control situation in which decisions were made individually and obviously no negotiation was involved. It also provides a basis for systematic conflicts detection. This was then followed by a negotiation to achieve a group decision. When the consensus was achieved or the time limit ended, the decision forms were collected and the experiments' post mortem was deployed. In the post mortem session, feedback from the participants was gathered to learn if the experiments were successful and to note weaknesses, if any, for future references.

4.3 Threats to Validity

First the 'silent objective' was defined as in Section 3.2. The participants should not have been aware of the aims and measurement being employed. The purpose was to hide the desired outcome of the experiments which might have influence the participants' decisions. This is usually known as "blind experiments" to prevent participants' expectations from influencing the results [52]. On top of this, the variables which were identified to be measured in every experiment such as the agreement level and the quality values were unknown to the participants. The silent objective was enforced to let the participants focus only on exercising negotiation in order to achieve group decision without considering the variables to be measured from the output.

Second was the double measurement for the requirements quality. In a series of experiments to measure the quality of requirements, the requirements produced by the negotiation effort were discussed, tested, analysed and proven twice. Two types of methods and measurements were deployed separately with different groups of participants; and yet produced similar result that is improvement in quality. The double measurement was seen to give a redundant check and to support one method with another.

Third is the blind marking. Kitchenham et al [52] stated that a researcher's enthusiasm for their own work may bias the trial. Therefore, a third party was involved to assist the researcher to collect and to mark the experiment results purely based on the data collected. It was then analysed and measured by Cohen's kappa. Cohen's Kappa [53] is an index of inter-rater reliability that is commonly used to measure the level of agreement between two sets of dichotomous ratings or scores. The measurement involved an independent statistician, who ensured that the results were represented and reported correctly.

5 Conclusion

It is crucial to ensure that the process of empirical investigation is carefully designed. This is to guarantee that the data being collected is reliable to support the underlying theory. Therefore, a framework which, consist of the identification of population and participants, the flow of experimental process and methods to minimize bias to the results is developed. Besides that, since negotiation effort is seldom applied during the requirements engineering process, the relevance and the effectiveness of negotiation is elaborated. A high level model is presented here and the negotiation process is elaborated. The underlying theory consists of the definition of term used in the research and the rationale of the usage is clearly justified. Lastly, the elements in the negotiation process which is motivated by several current researches are clearly stated and discussed.

References

1. Ahmad, S.: Understanding Requirements Engineering. In: International Conference on Engineering and ICT, Melaka, Malaysia (2007)
2. Damian, D.E.H., Zowghi, D.: The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization. In: IEEE Joint International Conference on Requirements Engineering. IEEE Computer Society, Germany (2002)
3. Zowghi, D., Coulin, C.: Requirements Elicitation: A Survey of Techniques, Approaches and Tools. In: Aurum, A., Wohlin, C. (eds.) Engineering and Managing Software Requirements, pp. 19–41. Springer, Berlin (2005)
4. Grunbacher, P., Syeff, N.: Requirements Negotiation. In: Aurum, A., Wohlin, C. (eds.) Engineering and Managing Software Requirements, pp. 143–158. Springer, Berlin (2005)
5. Al-Karaghoul, W., AlShawi, S., Fitzgerald, G.: Negotiating and Understanding Information Systems Requirements: The Use of Set Diagrams. *Requirements Engineering* 5, 93–102 (2000)
6. Boehm, B., Egyed, A.: Software Requirements Negotiation: Some Lessons Learned. In: 20th International Conference on Software Engineering, IEEE Computer Society, Kyoto (1998)
7. Damian, D.E.H., Zowghi, D.: An insight into the interplay between culture, conflict and distance in globally distributed requirements negotiations. In: 36th Annual Hawaii on International Conference, Big Island, Hawaii (2003)
8. Davis, A.M.: Just Enough Requirements Management. Dorset House, New York (2005)
9. Grunbacher, P., Briggs, R.O.: Surfacing Tacit Knowledge in Requirements Negotiation: Experiences using EasyWinWin. In: 34th Hawaii International Conference on System Science, vol. 1, p. 1062. IEEE Computer Society, Hawaii (2001)
10. Mohan, K., Ramesh, B.: Traceability-based knowledge integration in group decision and negotiation activities. *Decision Support Systems* 43, 968–989 (2007)
11. Nuseibeh, B., Easterbrook, S.: Requirements engineering: A Roadmap. In: Conference on The Future of Software Engineering, pp. 35–46. ACM Press, Limerick (2000)
12. Damian, D.E.H.: Challenges in Requirements Engineering. Computer Science Technical Report. The University of Calgary, Calgary (2000)

13. Grunbacher, P., Boehm, B.: EasyWinWin: A Groupware-Supported Methodology for Requirements Negotiation. In: 8th European Software Engineering Conference held jointly with 9th ACM SIGSOFT International Symposium on Foundations of Software Engineering, vol. 26, Toronto, Canada (2001)
14. Eged, A., Grunbacher, P.: Identifying Requirements Conflicts and Cooperation: How Quality Attributes and Automated Traceability Can Help. *IEEE Softw.* 21, 50–58 (2004)
15. Kaiya, H., Shinbara, D., Kawano, J., Saeki, M.: Improving the detection of requirements discordances among stakeholders. *Requirements Engineering* 10, 289–303 (2005)
16. Boehm, B., Hoh, I.: Conflict Analysis and Negotiation Aids for Cost-Quality Requirements. *Software Quality Professional* 1, 38–50 (1999)
17. Boehm, B., Hoh, I.: Identifying Quality-Requirement Conflicts. *IEEE Softw.* 13, 25–35 (1996)
18. Hans, W.N., Manfred, A.J., Matthias, J., Georg, V.Z., Harald, H.: Managing Multiple Requirements Perspectives with Metamodels. *IEEE Softw.* 13, 37–48 (1996)
19. Nuseibeh, B.: Conflicting Requirements: When the customer is not always right. *Requirements Engineering* 1, 70–71 (1996)
20. Curtis, B., Krasner, H., Iscoe, N.: A field study of the software design process for large systems. *Communications of the ACM* 31, 1268–1287 (1988)
21. Nuseibeh, B., Kramer, J., Finkelstein, A.: ViewPoints: meaningful relationships are difficult? In: Kramer, J. (ed.) *Proceedings of the 25th International Conference on Software Engineering 2003*, pp. 676–681 (2003)
22. Lee, M., Boehm, B.: The WinWin Requirements Negotiation System: A Model-Driven Approach. Vol. 2008. *citeSeer* (1996)
23. Boehm, B., Bose, P., Horowitz, E., Ming-June, L.: Software requirements as negotiated win conditions. In: *Proceedings of the First International Conference on Requirements Engineering 1994*, pp. 74–83 (1994)
24. Darke, P., Shanks, G.: Stakeholder viewpoints in requirements definition: A framework for understanding viewpoint development approaches. *Requirements Engineering* 1, 88–105 (1996)
25. Hoh, P.I., Olson, D.: Requirements Negotiation Using Multi-Criteria Preference Analysis. *Universal Computer Science* 10, 306–325 (2004)
26. Robinson, W.N., Volkov, V.: Supporting the Negotiation Life Cycle. *Communication of ACM* 41, 95–102 (1998)
27. Price, J., Cybulski, J.: L.: The Importance of IS Stakeholder Perspectives and Perceptions to Requirements Negotiation. *Australian Workshop on Requirements Engineering*, Adelaide (2006)
28. Kaiya, H., Horai, H., Saeki, M.: AGORA: Attributed Goal-Oriented Requirements Analysis Method. In: *IEEE Joint International Conference on Requirements Engineering*, Essen, Germany, pp. 13–22 (2002)
29. Robinson, W.N., Volkov, V.: Conflict-Oriented Requirements Restructuring. GSU CIS working paper. Georgia State University, Atlanta (1996)
30. Atkinson, R.: Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria. *International Journal of Project Management* 17, 337–342 (1999)
31. Boehm, B.: *Software Engineering Economics*. Prentice Hall, New Jersey (1981)
32. Smith, M.: A Small Experiment in International Negotiations: Chuo Law School. Japan and Chulalongkorn Law Faculty, Thailand. vol. 19, 216–220 (2005)

33. Hatton, S.: Software Requirements Prioritisation: The Client's Perspectives. Fifteenth University of Western Australia, School of Computer Science & Software Engineering Research Conference. CSSE, University of Western Australia, Yanchep, Western Australia, 51-62 (2006)
34. Karlsson, J., Ryan, K.: Supporting the Selection of Software Requirements. In: Proceedings of the 8th International Workshop on Software Specification and Design. IEEE Computer Society, Los Alamitos (1996)
35. Mohammed, S., Ringseis, E.: Cognitive Diversity and Consensus in Group Decision Making: The Role of Inputs, Processes, and Outcomes. *Organizational Behavior and Human Decision Processes* 85, 310–335 (2001)
36. Choudhury, A.K., Shankar, R., Tiwari, M.K.: Consensus-based intelligent group decision-making model for the selection of advanced technology. *Decision Support Systems* 42, 1776–1799 (2006)
37. Price, J., Cybulski, J.: Consensus Making in Requirements Negotiation: The Communication Perspective. *Australasian Journal of Information Systems* 13, 209–224 (2005)
38. Sommerville, I.: *Software Engineering*, 7th edn. Addison-Wesley, U.S (2004)
39. McConnell, S.: *Rapid Development: Taming Wild Software Schedules*. Microsoft Press, Redmond (1996)
40. Humphrey, W.S.: *Managing the software process*. Addison-Wesley Longman Publishing Co., Inc., Amsterdam (1989)
41. Christensen, M.J., Thayer, R.H.: *The Project Manager's Guide to Software Engineering's Best Practices*. IEEE Computer Society Press, Los Alamitos (2002)
42. Pressman, R.S.: *Software Engineering A Practitioner's Approach*, 6th edn. McGraw Hill, New York (2005)
43. IEEE Standard Glossary for Software Engineering Terminology. IEEE Standard 610.12-1990. IEEE Computer Society (1990)
44. *Software Engineering—Guide to the Software Engineering Body of Knowledge (SWEBOK)*. Standards Australia (2007)
45. Lauesen, S., Vinter, O.: Preventing Requirement Defects. In: Sixth International Workshop on Requirements (REFSQ 2000), Stockholm (2000)
46. Biffel, S., Freimut, B., Laitenberger, O.: Investigating the cost-effectiveness of reinspections in software development. In: Proceedings of the 23rd International Conference on Software Engineering, IEEE Computer Society, Toronto (2001)
47. Biffel, S., Halling, M.: Investigating the defect detection effectiveness and cost benefit of nominal inspection teams. *IEEE Transactions on Software Engineering* 29, 385–397 (2003)
48. Halling, M., Biffel, S., Grünbacher, P.: An economic approach for improving requirements negotiation models with inspection. *Requirements Engineering* 8, 236-247 (2003)
49. MoSCoW Prioritisation. *Reducing Your Acceptance Testing Risk*. Coley Consulting (2007)
50. Linstone, H.A., Turoff, M.: *The Delphi Method: Techniques and Applications*. Addison-Wesley Pub. Co., Advanced Book Program (1975)
51. Hsu, C.C., Sandford, B.A.: The Delphi Technique: Making Sense Of Consensus. *Practical Assessment, Research & Evaluation* 12 (2007)
52. Kitchenham, B., Pfleeger, S.L., Pickard, L.M., Jones, P.W., Hoaglin, D.C., Emam, K.E., Rosenberg, J.: Preliminary guidelines for empirical research in software engineering. *IEEE Trans. Softw. Eng.* 28, 721–734 (2002)
53. Cohen, J.: A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20, 37–46 (1960)

A Study of Tracing and Writing Performance of Novice Students in Introductory Programming

Affandy¹, Nanna Suryana Herman¹, Sazilah Binti Salam¹,
and Edi Noersasongko²

¹ Faculty of Information and Communication Technology, Universiti Teknikal Malaysia - Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia

² Faculty of Computer Science, University of Dian Nuswantoro,
Nakula I / No.5-11 Semarang 50131, Indonesia
{affandy_ra, nsuryana, sazilah}@utem.edu.my,
edi-nur@dosen.dinus.ac.id

Abstract. The tasks of programming include complex knowledge and skills that is, from understanding problems to evaluating validity of program. Novice students often face difficulties in learning programming due to various issues and the nature of the subject, which can be vague and invisible. A survey was conducted on 294 students from two universities to study novices' problems in dealing with tracking the logical flow and writing a simple code. The average score for tracking and writing skills were quite disappointing. Students were only able to master the static part of programming knowledge. They lacked the knowledge in understanding and tracing the dynamic behavior of the program. This research attempts to propose a model to shift the internal working memory load of students through integrated visualization tools that can reveal the dynamic behavior of programs and related concepts that appear in each level of program abstractions.

Keywords: learning, programming, novices, tracing, reading, writing, program visualization, behavior.

1 Introduction

The tasks of programming include complex knowledge and skills that range from understanding problems, designing problem-solving, constructing code, to evaluating validity of programs. Students need to obtain many computer subjects to understand programming as a whole e.g. Fundamental Programming, Data Structures, Algorithms and Problem-Solving, Event-Driven Programming etc.[1]. Generally, computer students seem to have problems in learning programming due to various issues and the nature of the subject, which is normally vague or even invisible. Furthermore, this situation leads to poor results in learning programming and a high dropout rate on the introductory of programming courses [2]. Previous studies by an ITiCSE working group in 2001 concluded a very surprising result that the average score of programming test of 217 students was only 22.9 out of 110 [3]. The further research in 2004 remained the same, only 27% of the 556 multi-national students achieved a higher score (10-12 of 12) of reading and tracing simple existing code [4].

In order to solve these programming education problems, some programming instructors use *Software Visualization (SV)* tools that are intended to represent information about software in a graphical way. There is a belief among the instructors that SV will give students a better understanding of basic programming because they can visually reveal the program's process and the inner workings of the algorithm. However, things have happened beyond that belief. Other studies show a downfall of this technology as a learning or teaching aid-tool [5,6]. Previous studies conclude that time and effort required to design, integrate and maintain the visualizations in the class have become the main obstacles for students and instructors to use it as a learning-aid tool [7,8]. Most of them provide exposure to the dynamic or static behavior of an existing program or *canned-algorithm* to support the understanding of student's knowledge of the program. Meanwhile, students rely on the program developer tools (e.g. C++, Turbo Pascal, VB) to train their skill in writing a computer program due to the lack of learning-aid tool that can support their skill in constructing a program effectively. Therefore, they understand the higher level of a particular algorithm but it is so hard for them to turn it into a lower level abstraction in a form of program code.

This study attempts to reveal novice students' performance in understanding the existing code and in writing a simple C++ program by the end of the first semester. This consideration will lead us to answer following questions: currently, what is the students' level of ability to trace and write the simple C program? Do they have the same level for both these capabilities? Furthermore, the research also attempt to propose a better approach to support students' program comprehension as a whole and to find a way to know how these needs can be met through a visualization tool for introductory programming courses.

2 Programming and Program Comprehension Process

Programming knowledge area includes essential skills and concepts that include investigating the problem, designing the problem-solving, transforming design into code and data structure by writing a highly constrained language, and verifying the validity of the program. Both practical and conceptual sides should be studied simultaneously. These synergetic approaches are aggregated further and should be maintained until a higher-level of program comprehension is achieved. Meanwhile, program comprehension is a process of reconstructing the programming knowledge that uses existing general and software specific knowledge in order to meet the ultimate goal of a code cognition task [9]. Some experts state diverse theories of program understanding process or cognition models but they share similar components of the mental model, a current internal developer's mental representation of the program to be understood, such as text structure, chunks, plans, and hypotheses but different in sequence of assimilation process, either a top-down, bottom-up or even merge both of them [9-11]. However, there is a similar requirement skill in the process of assimilation i.e. abstraction and translation ability, such as translating a word problem into sub-problems, abstracting the proper solution, translating solution into specific code and abstracting the behavior of the code.

Mostly conventional introductory programming courses have been delivered with an approach that resembles top-down and bottom-up strategy in separate manners. Instructors usually teach in an order of sequence. The broad conceptual framework of a particular programming, algorithm, problem-solving design and language structure is delivered first to students in a lecture class (top-down model). After that, through their limited knowledge of programming concepts coupled with complexity of highly constraint syntaxes, students struggle in an attempt to build programming code in accordance with designed algorithm (bottom-up model). This is contrary to the experiment shown [11] that elucidates the fact that novices jump from top-bottom model into bottom-up model and jump-back again in to top-bottom model arbitrarily in order to make a model and to correct the program. This cross-referencing work, between higher and lower abstraction of program, has lacks supported by instructors and developer of SV tools as a part of their features.

Currently the structure of introductory programming courses are based on lectures and practical laboratory work, which focus largely on knowledge of the language and building skills needed to generate a program. Instructors rarely teach how to verify the validity of a program in the evaluation phase, the facts that the interrelated tasks in designing algorithm, constructing program, and evaluating among them has been assumed as a sequence relationship rather than interwoven relationship. Many of them believe that the ability to write the code would be followed by the ability to evaluate/debug the code. Meanwhile Robins et.al [12] in their review of learning and teaching state that learning to make a program not only involves learning to develop the model of the problem domain and the desired program but also developing the tracking and debugging skill to model and maintain their hypothesis of their own program.

3 Survey Design and Methodology

In order to get some answers about the level of ability to trace and to write computer programs, students from two neighboring countries, University of Teknikal Malaysia Melaka (*T*) and Dian Nuswantoro University, Indonesia (*D*) were tested. First, students were tested on their ability to understand the knowledge inside and to trace the outcome of a short piece of program. Secondly, students were tested on their ability to write a simple C++ program based on specific requirements.

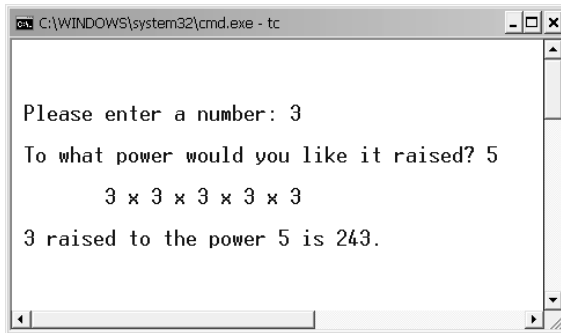
3.1 Participants

A survey was conducted on 294 undergraduates from both universities who took an introductory programming subject in the first semester ($n_T = 140$, $n_D=154$). Some students from the non-IT study programme too, responded to the survey question.

3.2 Material and Task

The questionnaire materials included multiple-choice questions with single answers based on reading or tracing capability test, and writing a simple C program test. The

student's performance in program comprehension was evaluated by using two measurements, tracing and writing program test. The tracing ability test examined comprehension of following program knowledge: *elementary operation* (P_1), *control-flow* (P_2), *data-flow* (P_3), *program function* (P_4), and *program state* (P_5), which was developed by Pennington in order to study differences in programmers' comprehension strategies [13]. Each category consists of a short simple of C++ program followed by three multiple-choice questions related to the program and the maximum score for this section is 15



```

C:\WINDOWS\system32\cmd.exe - tc

Please enter a number: 3
To what power would you like it raised? 5
      3 x 3 x 3 x 3 x 3
3 raised to the power 5 is 243.
  
```

Fig. 1. The expected output display of the writing-code test from case of a math operator for raising an input number to a particular power

The writing-code test was a paper-based test to assess participants in developing a simple C++ code based on specific requirements. The scenario of the problem was students were requested to construct a simple code that raises any number of X to a positive power of N (X and N are data-input from keyboard), the output will display the number of X for N times, and the results of power operation, as seen on Fig. 1. The test was developed by adopting fundamental computer process (*input, process, and output*) into following categories of skills: *data definition* (S_1), *input session* (S_2), *control-flow* (S_3), and *output session* (S_4). One point will be awarded to the participant for each valid line code referred to each in category, and the maximum score for this section is 14 point.

3.3 Procedure

Survey was conducted based on an individual paper-pen test in limited time and students were prohibited using computers and books to help them in answering the questions. For program comprehension test, each section (tracing and writing program test) was allocated thirty minutes to complete the answer. The test was conducted between week#13 and week #14 of a 14-week course.

4 Performance and Data Analysis

This section contains a statistical analysis of the two performance data that are provided by the students from the different universities. Some of the dataset analyzed and presented either as independent or combined dataset.

4.1 Analysis of Tracing Ability Score

The average score of total tracing/reading ability (P_1 - P_5) for all students, all exercises, at both universities were 7.57 out of 15 (stdev: 2.44). The score for each university is generally similar as in Table 1.

Table 1. Tracing/Reading Ability

University	Average	Stdev
T (n = 140)	7.83	2.30
D (n = 154)	7.33	2.54

Table 2. Average Score of Trace/Read Code by Trace Tasks

Trace task	Average (stdev)			Sig. (2-tailed) <i>p</i>
	Univ D	Univ T	Combined	
P_1	1.80 (0.96)	1.77 (1.01)	1.81 (0.98)	0.793
P_2	1.33 (0.96)	1.73 (0.89)	1.51 (0.94)	0.001
P_3	1.26 (0.83)	1.40 (0.75)	1.32 (0.80)	0.15
P_4	1.48 (0.71)	1.47 (0.83)	1.48 (0.77)	0.937
P_5	1.46 (1.03)	1.45 (0.90)	1.45 (0.96)	0.911

Even though there are some differences in the teaching and learning process between students at both universities, we assume that it is possible to combine data from both the universities. We used a paired-sample t-test to compare the similarity on each of the trace-code task, and its result shows that almost all tasks of trace/read test do not differ significantly ($p > 0.005$), and only P_2 differs significantly ($p = 0.001$). Fig. 2(a) shows that visually the distribution of trace score approaches to a normal distribution, but based on *one-sample Kolmogorov-Smirnov test* we find that the 2-tailed significance of the test statistic is very small, 0.006, meaning that tracing ability may not be assumed to come from a normal distribution with the given means and standard deviation. Fig. 2(b) also confirms that students do the best only on elementary operation task and four other tasks remaining below 1.5 out of 3.

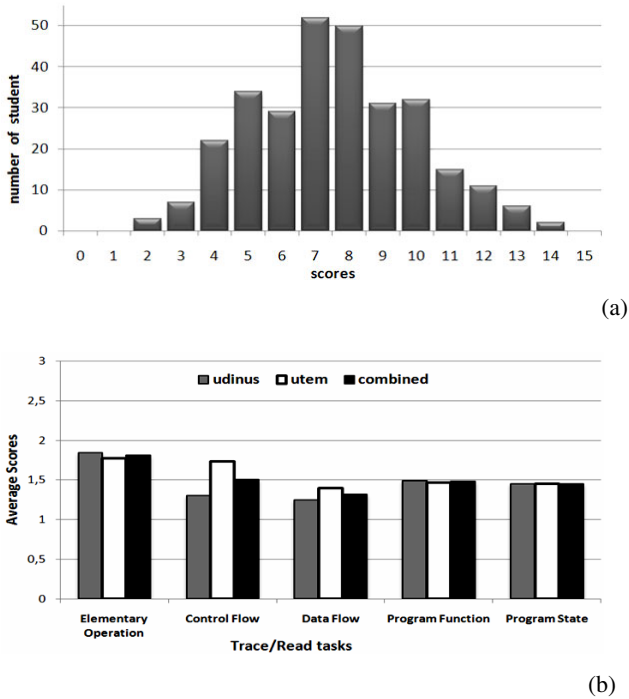


Fig. 2. (a) Tracing ability score distribution. (b) Average score of trace tasks

4.2 Analysis of Writing-Code Ability Score

In general, a worse situation occurred also on the test of ability to write a program. Students’ performances in both the universities are very low compared to the maximum score. Based on the fact that most scores of each task differ significantly to each other ($p < 0.05$) then we cannot assume a combined data from both of them (see Table 3). Students did the best in declaring input statement (S_2), and next on defining data type (S_1). This may be due to the simplicity of the standard input statement and the basic data type at the introductory level. Whereas, students’ skill in the application of control statement is very low, perhaps due to their lack of understanding on control flow (P_2) and data flow (P_3).

Table 3. Average Score of the Writing-Code Test

Writing tasks (max score)	Average		Sig. (2-tailed) P
	Univ D	Univ T	
S_1 (4)	2.28 (1.16)	1.31 (1.38)	0.000
S_2 (4)	2.40 (1.82)	2.20 (1.81)	0.298
S_3 (4)	0.38 (0.89)	0.11 (0.41)	0.001
S_4 (2)	0.54 (0.74)	0.30 (0.53)	0.001
S_1-S_4 (14)	5.59 (3.79)	3.91 (3.51)	0.000

Fig. 3(a) shows that the distribution of the writing-test score spreads along the scores axis, while the majority of the students did very poorly, more than 60% of the students got below 7. There are some little “humps” in the distribution, indicating that very few students with somewhat better score, less than 35% of the students scored above 8. Many students at the university *T* left their answer sheet blank for some reason, e.g. insufficient time, shortage of computers to work on, not knowing what should be written first since not accessible to template of program etc.

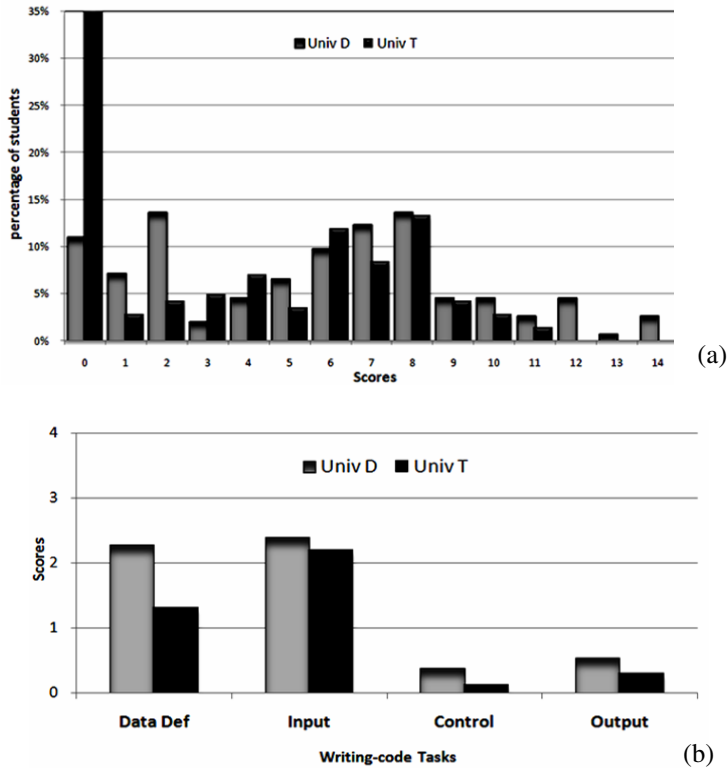


Fig. 3. (a) Writing-code score distribution (b) Average score of the writing-code tasks

5 Discussion

It was admitted that problem-solving skill as well the ability to read or trace a program contributes to the programming skill and further to the program comprehension. According to the mental model approach, during program comprehension, program model should be developed first before the situation model. Program model is constructed by using a combination of static elements and dynamic behavior elements of the mental model. Once the program model representation exists then the situation model is developed.

5.1 Ability to Trace the Lower-Level Abstraction

It seems that during the development of the program model, participants can only possess static elements of the mental model that are micro-structure and macro-structure of the program text [9]. Micro-structure consists of actual program statements and their relationship e.g. *variable definition*, *elementary operations*, *sequence*, *loop constructs* and *conditional constructs*. Whereas macro-structure is identified by a label or procedure name that corresponds to control-flow organization of the program text, e.g. *program function*, *program state* and *module label*. Since the static element is related to program text and its structure, it is easier for students to comprehend it, as in figure 2(b) that all these elements can be understood equally. The experience in programming improves the understanding level of text-structure knowledge and store in long-term memory. The role of *beacon*, a guiding text for gaining a high-level of understanding (e.g. variable name, function name, literal string, etc.) and *rule of discourse*, conventions of coding,[14] within the trace-code test may also contribute to the comprehension.

5.2 Inability to Construct the Higher-Level Abstraction

Unfortunately, things that occur in the understanding of the program text do not happen with the acquisition of dynamic behaviors of program text as an integrated element of the mental model. Respondents fail in implementing both mechanisms that produce information, *chunking* and *cross-referencing*. Chunking means taking several *chunks* of lower-level structure to create new higher-level-abstraction structure [15]. For instance, a piece of code may represent a *compound if-conditional*, this section of code takes a job from received input value, evaluates the value, and then saves the value into a related array variable leading to the higher-level-abstraction of “*Categorization Process*”.

Based on table 2 and table 3, respondents can answer the given questions related to *if-conditional* in the trace code test (mean score was 1.51 out of 3), but they are unable to chunk this knowledge structure to construct the *if-conditional* within the write code test (mean score in each university is $S_3 = 0.11$ and 0.38 out of 4). In the write code test, micro-structure of *if-conditional* should be *chunking* as macro-structure of “*Print Evaluator*” to evaluate whether the character “*x*” (as a multiply symbol) will be displayed or not. Moreover, this situation leads to the lack of ability to *cross-reference* which relates to different levels of abstraction.

5.3 The Need for Multi-representation of Program

Program representation is a very complex multidimensional representation, as Soloway states [14] that there are two audiences of computer program, the computer and human reader. For the human reader, computer program can appear in several level of cognitive representation:

- Level 1, text-structure representation
- Level 2, control-flow representation
- Level 3, functional representation
- Level 4, problem domain representation

Level 1 corresponds to the lower-level abstraction of the program, which is in the form of highly constraint syntaxes. Level 2 and 3 refer to intermediate-level abstraction, they reflect *how* it works (mechanism) and *what* the end result is. Level 4 corresponds to the higher-level abstraction of the code in terms of real world problem.

Comprehension process becomes more difficult since students have to maintain all those representations while keeping in their mind all of the cross-referencing between designed algorithm, syntaxes of particular language programming and also its semantic result at the same time. Because of those overloading information to depict program abstraction, the use of software visualization tools are expected to relieve the internal working memory load by making the meaning of the code more apparent and concrete, while making the overall structure of the program easier to grasp. Unfortunately, the involvement of these technologies that are used to assist in comprehension tasks cannot immediately resolve the program comprehension problems. Some visualization tools that were released around 1999 to 2009 such as Alice 3D [16], ALVIS Live! [17], ANIMAL [18], DataStructure Navigator [19], Data Structure Visualization [20], Jeliot 3[21], MatrixPro [22], Raptor [23], The Teaching Machine[24], ViLLE [25], these tools mostly support one or two aspects of the programming representation either algorithm, data structure, or program visualization rather than bridging multi-representation of the program. Results of the survey confirm that students can understand the program code in the form of text-structure and they can clearly understand the problems that are represent in natural language. The biggest challenge for them is the in-between representation, which relates to the lower and higher level of abstraction.

5.4 Proposed of Multi-representation Model

Novices find that it is difficult to map the problem domain into the functional representation and the graphic elements of the visualization to the syntax of the program. However, IT graduates must also realize the importance of abstraction; they must be able to manage the complexity of the programming through the abstraction [26]. These facts lead us to focus our attention on abstraction ability. It is believed that this ability has a contribution to the ability to trace and write a computer program. However, since learnt to think abstractly is very difficult, we propose a model that can help us to construct a learning-aid tool that will reveal or visualize the multi-representation of the program.

The basic idea of the model is to show novices about the programming stages starting from designing problem-solving, developing code and validating logical flow of the program through dynamic graphical view of visualization (see Fig. 4). At the first layer, model has a collection of packages of basic solutions that can be use to construct a bigger plan of problem-solving. Students can even modify or create their own basic solution from the scratch. The construction of several packages will create a specific function as a part of the bigger plan of problem-solving which can be seen gradually as a whole by students. The second layer of the model will demonstrate how each package works. It will show the logical process of the package and how the package will process the data. Generally, this layer will show to novices the changing states of the program that they never saw before by using a standard program developer. Model at the third level will transform the package into a textual structure

of specific programming language. Textual structure is built by applying several formatting techniques to create a programming code that looks like a descriptive document. This also applies conversation standard, which ease the novices to construct their hypothesis while they read program codes. All models are integrated in a learning-aid tool that will be helpful for students to comprehend the program from a higher-level to a lower-level since the process of constructing a certain program can be seen clearly e.g. how the process works, where the data come from, the results and the textual code of the program.

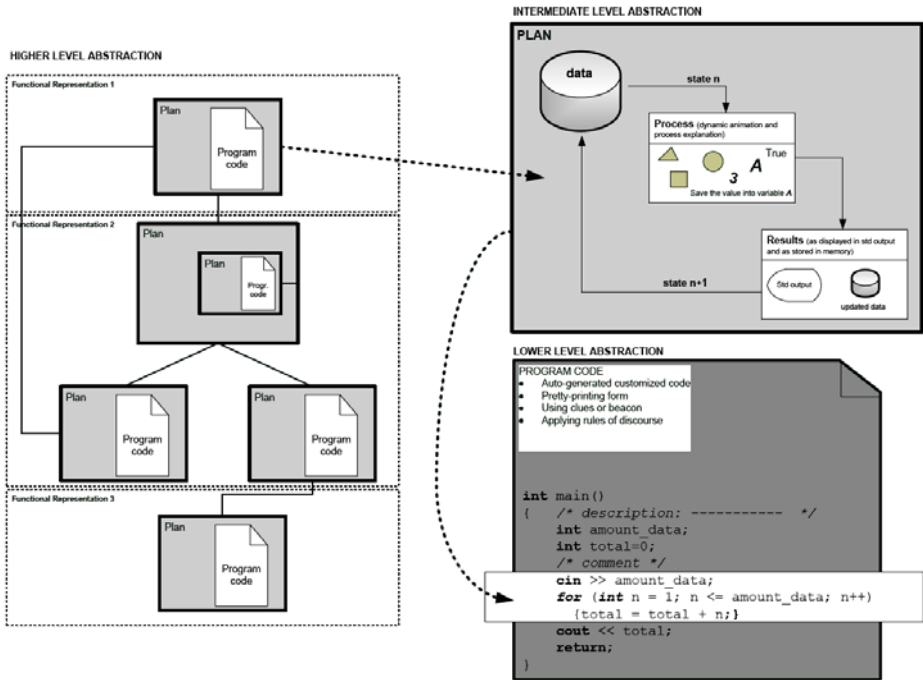


Fig. 4. Integrated Multi-Representation Model

Due to the diversity in the comprehension process, our proposed model is designed by adopting both the top-down and bottom up comprehension as theoretical underpinnings to develop a comprehension support tool [15]. Tool is modeled by following the point of view of the abstraction level of the program. For the top-down approach the abstraction view start from a higher-level model to a lower-level model and vice versa for the bottom-up approach. We distinguish the following three models:

1. **The higher-level abstraction model**, provides a static graphical and less-textual representation of algorithm. This model intends to show the structure and the logical flow of some functional representations of the program. The problem-solving design is divided into some functional representations using a modular approach in order to raise the level of abstraction and to enable students to achieve

and manage the complexity. Students using some packages of basic *plans* – stereotypical, canned solutions– provided by system or customized by the user, to develop a functional representation. *Plans* are symbolized with flowchart-like to reveal its logical flow. Combining some *plans* to achieve specific functions similar to installing a piece of puzzle to get a more meaningful picture. The environment is designed to foster active learning that engages students to construct their own algorithm using some basic *plans* in order to support the visual engagement at level 4 (constructing a visualization) [27] rather than providing *canned algorithm*.

2. **The intermediate-level abstraction model**, this model is also known as the explanation model that provides a dynamic representation of data state changes and program behavior. The model takes each *plan* from previous level as the input, and shows to student how the symbol is executed or evaluated. *Plans* that are currently under execution are highlighted on the prior model to synchronize user view between high-level and intermediate level abstraction. For instance, when the model executes a piece of *plan* called TESTING-INPUT-VALIDITY, e.g. while `input_value < limit` do, it will dynamically and systematically show the process, starting from taking the value from variable `input_value` and `limit` then followed by the comparison process between both of them. The model demonstrates to students how each part of the *plan* is being executed and where all the values shown come from or go to rather than allowing student to guess. As a result, the logical consequences are displayed and the effect of the result, whether **FALSE** or **TRUE**, is addressed by the higher-level abstraction model to execute the next corresponding symbol/*plan*.
3. **The lower-level abstraction model** provides textual representation in the form of particular programming language. It transforms each *plan* of the higher-level abstraction model into syntax of the specific language right after the user constructs or edits a *plan*. The executed plan in intermediate and higher-level model is highlighted to show the mapping process among all levels. A *plan* can be constructed by one or more related statements that form a block of statement. Model intends to show clearly in terms of logic as to how a code is built line by line by referring to the correspondence of the designed algorithm at a higher-level abstraction. It also emphasizes the application of the *beacon* and *rules of programming* in order to gain a higher level of understanding. *Beacons* act as cues to the presence of certain structures or features that possibly lead to the creation of hypotheses e.g. procedures and variable names. *Rules of programming* are rules or conventions within programming such as code presentation or naming standards [9]. As well as in terms of formatting, code is presented with beautification in order to make it easier for reading and understanding. Code beautification involves parsing the textual source code into proper code formatting via the use of indentation, positioning of braces, blocking, coloring reserved words, size, and styling.

For instance, when the averaging problem is given to the student, they initiate by transform problem into sub-solutions such as entering the data, accumulating the data, counting the number of data, dividing the total accumulation of data with the number of data and etc. These sub-solutions will be represented as higher-level abstraction

and student will construct each of sub-solution using provided plans or create their own plans. For example the `SimpleType_DataDefinition` plan and `SentinelControlled_Looping_DataInput` plan will construct a function of entering the data. Within intermediate model `SimpleType_DataDefinition` plan will be visualized by dynamic animation of artifacts that show the process of data definition. In the same way to show where all of each data go to and when the iteration will stop, the model will execute and animate the `SentinelControlled_Looping_DataInput` plan right after previous plan. At the same time textual representation model will generate the corresponded statement automatically for each plan under execution.

It is an integrated and reversible model, so when students try to trace their own designed algorithm, the higher-level model shows the complete logical flow of the algorithm and at the same time, the execution model allows students to evaluate the behavior of their algorithm in relation to changes of data value and flow of program. Meanwhile the lower-level model helps students to map the higher-level abstraction into lines of code. It can be said that the concepts that appear in each model are explicitly related to each other.

6 Conclusion

There exists a mutual and complex dependency between understanding the conceptual framework of algorithm and the ability to construct and to debug a program. The ability to trace a program becomes one of the factors that are related to the ability to solve problems, and the ability of problem-solving contributes to the programming skill. Since program is invisible, learning to make the program requires an effort to make an abstract of each programming element. Integrated program visualization as learning-aid tool is needed to shift the internal working memory load of students to provide more “*space*” to the essential knowledge of programming. Integrated means that tool should reveal the structure of higher-level abstraction of the program, process, and behavior of the program, and gradually the construction program in a particular syntax of programming language.

Development of learning-aid tool with such complexity can be used to help students who have different learning strategies to understand the essentials of programming. This can be a greater challenge and can be studied in future research. Currently we are in the progress of developing the prototype based on our model which is hoped to shed light on the program understanding of our novices students

Acknowledgments

This work is done with support from the University of Teknikal Malaysia, Melaka via the short term grant projects PJP / 2010 / FTMK (12E) S720, in collaboration scholarship from the University of Dian Nuswantoro, Indonesia.

References

1. Association for Computing Machinery.:Curriculum and guidelines for undergraduate degree programs in information technology. (ACM) - IEEE Computer Society (2008)
2. Paivi, K., Malmi, L.: Why Students Drop Out CS1 Course? In: ICER 2006 - 2nd International Computing Education Research Workshop, pp. 97–108. ACM, New York (2006)
3. McCracken, M., Almstrum, V., Diaz, D., Thomas, L., Guzdial, M., Utting, I., Hagan, D.: A multi-national, multi-institutional study of assessment of programming skills of first-year CS students A framework for first-year learning objectives. *ACM SIGCSE Bulletin* 33, 125–180 (2001)
4. Lister, R., Seppälä, O., Simon, B., Thomas, L., Adams, E.S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., McCartney, R., Moström, J.E., Sanders, K.: A multi-national study of reading and tracing skills in novice programmers. *ACM SIGCSE Bulletin* 36, 119 (2004)
5. Hundhausen, C.D., Douglas, S.A., Stasko, J.T.: A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages & Computing* 13, 259–290 (2002)
6. Ihanntola, P., Karavirta, V., Korhonen, A., Nikander, J.: Taxonomy of effortless creation of algorithm visualizations. In: ICER 2005 - International Workshop on Computing Education Research, pp. 123–133. ACM Press, New York (2005)
7. Levy, R.B.B., Ben-Ari, M.: We work so hard and they don't use it: acceptance of software tools by teachers. *ACM SIGCSE Bulletin* 39, 250 (2007)
8. Cliburn, D.C.: Student opinions of Alice in CS1. In: 38th Annual Frontiers in Education Conference, FIE 2008, p. T3B–1. IEEE, New York (2008)
9. Maryhauser, A.V., Vans, A.M.: Program Understanding - A Survey. Technical Report, Colorado State University (1994)
10. Pennington, N.: Stimulus structures and mental representations in expert comprehension of computer programs. *Cognitive psychology* 19, 295–341 (1987)
11. von Mayrhauser, A., Vans, A.M.: From code understanding needs to reverse engineering tool capabilities. In: Proceedings of 6th International Workshop on Computer-Aided Software Engineering, pp. 230–239. IEEE Comput. Soc. Press, New Jersey (1993)
12. Robins, A., Rountree, J., Rountree, N.: Learning and teaching programming: A review and discussion. *Computer Science Education* 13, 137–172 (2003)
13. Pennington, N.: Comprehension strategies in programming. Ablex Publishing Corp., New York (1987)
14. Soloway, E.: Learning to program = learning to construct mechanisms and explanations. *Communications of the ACM* 29, 850–858 (1986)
15. Storey, M.-A.: Theories, tools and research methods in program comprehension: past, present and future. *Software Quality Control* 14, 187–208 (2006)
16. Cooper, S., Dann, W., Pausch, R.: Alice: a 3-D tool for introductory programming concepts. *Journal of Computing Sciences in Colleges* 15, 107–116 (2000)
17. Hundhausen, C., Brown, J.: What You See Is What You Code: A 'live' algorithm development and visualization environment for novice learners. *Journal of Visual Languages & Computing* 18, 22–47 (2007)
18. Robling, G., Schuler, M., Freisleben, B.: The ANIMAL Algorithm Animation Tool. In: The 5th Annual SIGCSE/SIGCUE ITiCSE Conference on Innovation and Technology in Computer Science Education, pp. 37–40 (2000)
19. DSN: Data Structure Navigator,
<http://dbs.mathematik.uni-marburg.de/research/projects/dsn/>

20. Data Structure Visualization,
<http://www.cs.usfca.edu/~galles/visualization/>
21. Moreno, A., Myller, N., Sutinen, E., Ben-Ari, M.: Visualizing programs with Jeliot 3. In: The Working Conference on Advanced Visual Interfaces - AVI 2004, p. 373. ACM Press, New York (2004)
22. Karavirta, V., Korhonen, A., Malmi, L., Stalnacke, K.: MatrixPro -a tool for demonstrating data structures and algorithms ex tempore. In: International Conference on Advanced Learning Technologies, pp. 892–893. IEEE, Los Alamitos (2004)
23. Carlisle, M.C., Wilson, T.A., Humphries, J.W., Hadfield, S.M.: RAPTOR: A Visual Programming Environment for Teaching Algorithmic Problem Solving. ACM SIGCSE Bulletin 37, 176 (2005)
24. Teaching Machine, <http://www.engr.mun.ca/~theo/TM/>
25. Rajala, T., Laakso, M.J., Kaila, E., Salakoski, VILLE-A, T.: language-independent program visualization tool. In: The Seventh Baltic Sea Conference on Computing Education Research (Koli Calling 2007), pp. 15–18. Australian Computer Society, Inc. (2007)
26. Interim Review Task Force.: Computer Science Curriculum 2008: An Interim Revision of CS 2001. Report from the Interim Review Task Association for Computing Machinery and IEEE Computer Society (2008)
27. Naps, T.L., Rodger, S., Velázquez-Iturbide, J.Á., Röbling, G., Almstrum, V., Dann, W., Fleischer, R., Hundhausen, C., Korhonen, A., Malmi, L., McNally, M.: Exploring the role of visualization and engagement in computer science education. ACM SIGCSE Bulletin 35 (2003)

A Review of Prominent Work on Agile Processes Software Process Improvement and Process Tailoring Practices

Rehan Akbar¹, Mohd Fadzil Hassan¹, and Azrai Abdullah²

¹ Department of Computer and Information Sciences

² Department of Management and Humanities

Universiti Teknologi PETRONAS, Malaysia

rehankb@yahoo.com, mfadzil_hassan@petronas.com.my,
azraia@petronas.com.my

Abstract. Global software development has changed the overall software development practices. It has introduced various new software development processes and methodologies. A new generation of processes has increasingly replaced the traditional software engineering practices. Emerging practices such as agile based methodologies, software process tailoring, process improvement and management approaches have gained much attention during the recent years. Software engineering researchers have produced a number of good quality works in such areas. In this paper, a review of major contributions of the researchers on various aspects of software development processes is presented. Specifically, the analysis on different approaches of process improvement and tailoring is critically discussed in the paper. This research provides guidelines to the researchers on future research directions. The research emphasizes on the need of industry oriented practical approaches of software development to meet the challenges of global software development.

Keywords: Agile, Globalization, Process Improvement, Management, Tailoring.

1 Introduction

The last decade of the twentieth century was the beginning of the advancements in Information Technology (IT). Most advanced tools and technologies for software development were introduced and utilized. Unlike, traditional heavy weight approaches of software development, the software companies started giving preferences to the new alternatives such as light weight agile based methodologies. Such drastic changes occurred as the consequences of IT globalization [1]. Not only IT, the globalization has also affected the overall international socio-economic fronts and political scenarios of the societies [2].

Since the start of the twenty-first century, the effect of IT globalization has become more intense. Information technology was among the fields which were greatly affected by its consequences. Project outsourcing and agile methodologies are

reported as two major products of IT globalization. Project outsourcing and factors involved in it were the basis of the increase in the use of the agile practices. [3] has discussed the reasons behind this change. It has generally setup the new trends in the software development industry. Preferences of software development teams and their criteria of selecting suitable tools, technologies and processes for software development have been changed due to the wide range of available alternatives. Project outsourcing proved to be the advent of new generation of processes in software engineering [4]. It is believed that traditional software development approaches could not withstand with the newly emerging practices and gradually became outdated.

The consequences of globalization started to appear during 1990s. Till the late 1990s project outsourcing had become the most common practice in the software industry. Mostly projects were being outsourced to the offshore companies. Offshore development started a new debate among the researchers regarding the processes, project management and performance issues. Many researchers have written a lot about it for example [4], [5], [6], [7], [8], [9]. In 2002, [10] presented a conceptual model for offshore development as shown in figure 1. In the model effort, elapsed time and rework were introduced as three factors of project performance measurement. The effect of the variables like quality processes, technical processes and communication & coordination on performance was measured through empirical data.

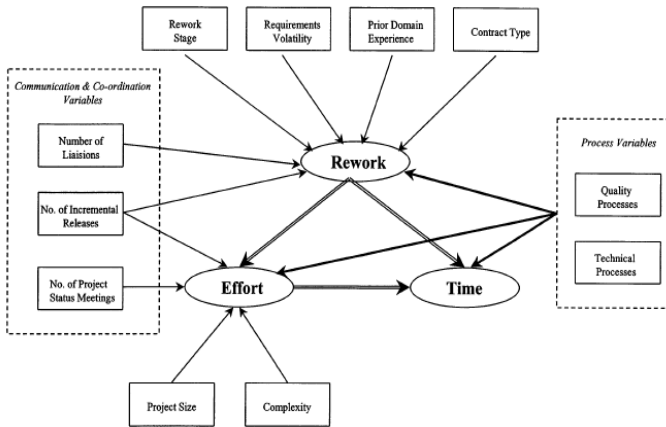


Fig. 1. Conceptual model

In another study [11] emphasized on the importance of the role of cross cultural issues in the performance of outsourced projects. Project outsourcing is also attributed to the geographically distributed development. In this regard, a road map to make a governance framework for distributed software development is presented in which characteristics of organizational level standardization, project execution, planning and infrastructure were discussed [12]. For small and medium scale organizations [6] described common issues and their solutions in outsourced projects

such as contract management, demand supply management, documentation, tool support, cultural and team level. These approaches provide significant control over such issues.

In short, as the consequences of IT globalization, a paradigm shift from the traditional heavy weight software development practices to the light weight agile based methodologies has been reported by many researchers. Since late 1990s till to-date the response to agile methodologies is overwhelming. The sole reason behind this was the style of development emerged as a result of outsourcing projects to offshore teams.

The role of software engineering research community has been very important in this regard. They have produced a good quality work, and have introduced the efficient light weight techniques of software development and project management. Agile methodologies, process tailoring, process improvement and management practices were the major areas in which most of the evolutionary work was produced during the last twelve years and further contributions are still being made.

In this paper we have presented the review of the contributory work produced by the researchers in all the three areas mentioned above. The review of the selected good quality papers is made in an analytical way. The papers were selected based on the novelty, key ideas and methodology used. Only a limited number of papers could fulfill these criteria. The agile practices are discussed in general, and a critical review on the process improvement and tailoring models, frameworks and standards is made in particular. Based on the analysis of the reviewed work, we have made conclusion on the future trends and practices of software development and research works.

The structure of the paper is as follows. In section 2, the prominent work on agile based methodologies is presented. Section 3 covers the review on process improvement and management approaches, while important models and frameworks of process tailoring are discussed in section 4. Conclusion is presented in section 5.

Finally, the paper suggests the solutions to the global software development challenges and suggests future research directions. This paper provides a hands-on review of past and current software development practices. It also critically analyzes the applicability and practicability of these approaches in accordance with the software development industry. Based on the analysis and overview, we have recommended the practices which are considered the best for the software industry. This paper emphasizes on the realization of the need of applied industry based solution oriented research works and software processes.

2 Agile Methodologies – A paradigm Shift

The clients of software projects prefer to launch their products early in the market to compete with their business rivals. This requirement of the clients keeps the developers under continuous pressure [13], [14]. To compete with the market, the ultimate requirement of the client is only the working code [4]. The support through fast paced development to release the working code early is provided in agile models [15]. Agile models emphasizes on minimum or no documentation. This aspect started debates on agile methodologies among the researchers. Two schools of thoughts in which one supports the agile models and the others to the traditional approaches [16],

[17] have born. Two agile based methodologies Extreme Programming (XP) [18], [19] and Scrum [20], [21] among others are widely used by the developers due to the available support of fast paced development. [16] has discussed agile methodologies in connection with ISO standards. Software engineering researchers have produced a number of models and frameworks on agile based methodologies. In 2001, agile manifesto comprising of twelve principles was formed in order to standardize the agile methodologies [15]. Like other approaches, agile methodologies also have some limitations with respect to the different environments and project requirements. In a study [15] has identified the following limitations of agile for distributed development, subcontracting, reusability, large teams, quality of safety critical products, and large and complex software systems.

The disadvantages and benefits of agile processes that are claimed by [15] are based on a set of assumptions. Therefore, their existence and non-existence in different environments may be doubted. As mentioned earlier that there is always a debate between agile supporters and traditional approaches supporters. In this regard [17] has compared traditional approaches with agile based mainly on the factors such as control, project management, team roles, way of communication and role of client. The control of project, project management and role of client are considered more important and critical in agile as compared to traditional approaches. Other researchers such as [22], [23], [24], [25], [26] have presented the similarities and differences between both approaches. As a matter of fact all software engineering methodologies have limitations [27]. In their framework (CHAPL) to understand the relationship between both approaches, [28] has concluded that traditional software engineering approaches and agile methodologies have “common philosophical origins” and are “technically compatible and complementary” to each others. This arises the need of a reasoning framework to determine the suitability and selection of software engineering methodologies in particular circumstances [27].

In addition to other factors, the selection of SE methodology also depends on the size of the company. Limitations of resources like financial, human etc forces the small companies to adopt light weight methodologies. Large companies with good resources pool prefer more standardized approaches. The web based application development has gain more popularity during recent years. Such kind of web development is being done in almost all the companies irrespective of their sizes, scales and environments. Unlike large organizations, small organizations face the situation of un-decidability many times during the project life cycles. In 2007, [29] discussed the software process models for web based application development in small software development companies. Light weight agile methodologies have proven to be the most result oriented methodologies for small, medium as well as large companies. In developing countries 75% - 80% companies are small and medium sized. Irrespective of the size of the company, agile methodologies are equally beneficial for them because the quick access to the ultimate goal of project success, which is similar for all the companies. Agile methodologies for example Extreme Programming (XP), crystal and Scrum have proved their worth in all kinds of environments. [30] has summarized an overview of XP and have made recommendations on how pair programming, a form of XP, can be implemented. In today's environment, the development time of web based applications has been reduced to few months. In current circumstances, the most important phase of

software development is requirement gathering, analysis and tracking. The agile approaches have also been quite efficient on this phase. In this regard, the agile hypertext design method has been proposed by [31]. In another work, [32] has proposed an agile approach for web systems engineering. In a South African empirical study of 59 projects, results show that agile practices are significantly beneficial in process improvement and project success which ultimately leads to the satisfaction of stakeholders [33].

As mentioned earlier by [27] that all software engineering methodologies have limitations, so there is always a margin for improvement. Software engineering researchers are continuously working on the process improvement and management practices. A number of models, frameworks and standards have been presented by the researchers in this regard. In the next section we have summarized some of the important contributions of the researchers on process improvement and management.

3 Process Improvement - Part of Process Management

IT globalization has made the software process improvement and management the key research areas in software engineering today. Each organization irrespective of its size, scale and types of the projects has to rely somehow on a software process for its development and management tasks. This dependency has made the development processes more critical element in the growth and success of projects.

The recent advancements in the field of IT have changed the overall software development scenarios. Organizational structures, preferences and priorities of the clients and the organizations have been changed. From hierarchical structures of departmental based divisions, organizations have shifted to the business process oriented team structures [34]. In a research work [34] has presented a reference framework for process-oriented software development organizations. According to the framework, management and support processes are considered necessary for the output of the project and client's satisfaction. A hierarchy to organize processes is proposed with the process management team at the top and process execution teams & their team leaders at the bottom in the hierarchy. The framework is then examined in the context of a generic as well as a software development organization. The framework provides the basic structure to establish process oriented organizations. Emphasis is given to the processes as critical factor and their continuous improvement and adaptation.

In another study, a review on the roles of the water fall model, capability maturity model (CMM), ISO-9000, SPICE, Trillium and BOOTSTRAP in process improvement are discussed by [35]. The water fall model is considered as the foundation model in organizing the software processes activities. Still this model is widely used in the companies. The Capability Maturity Model (CMM) provides various key process areas (KPA) for the process maturity, improvement and standardization. The problem with the CMM standardization approach is that it has limitations for small and medium scale organization due to scarcity of the resources. It seems more applicable for large organizations. ISO-9000 is a general quality standard and so far further IS standards have been introduced for software development such as ISO-9001. It is believed that process improvement and

management approaches may vary based on organizational goals, priorities and project requirements [35].

In 2004, [36] discussed process improvement approaches such as capability maturity model, six sigma, lean development and ISO-9001 and raised the issues such as :

- i. Lack of academic research factor on the efficiency and effectiveness of these approaches.
- ii. "They are based on very similar concept and techniques."
- iii. They are unable to present best practices specifically for the software development.
- iv. They are just the improvement approaches for established processes.

The gap between academic researchers and actual industry practitioners is believed to be the basic reason behind the ineffectiveness of various approaches [37]. The researchers don't find resources to collect the real data from the industry and on the other hand industry people also seem unwilling to cooperate with the researchers and consult their published work to find the solutions of the problems. This has led to a situation where software engineering research has been ineffective to meet the industry issues. Two limitations in software engineering research methodology namely poor understanding of the theory and inability to perform suitable testing have been reported by [38]. Unlike other fields of sciences, research methodologies in software engineering are not well established and structured. In 1995 and later, [39], [40] in their critical study on experimentation in software engineering found the software engineering papers worse among that of 43% papers of computer science in general with poor experimental design and testing methodologies. [41], [42] had also the same kind of observations. In 2002, [43] described the types of research questions, research strategies, types of results and validation techniques in software engineering research. Similarly many other researchers such as [44], [45], [46] have presented a good quality work on software engineering research methodologies. The areas of process improvement and management have always been preferred areas of research. Both areas have become more critical since the start of global software development.

With regards to the size of the organizations, process management and improvement practices have become more problematic and important. In this regard, [47] has proposed an integrated process management system for project management and business processes as a part of a workflow management. The emphasis is given on automation and tool oriented management systems. Enterprise level organizations usually adopt process improvement and management practices to get benefit from the business. Presumably process improvement practices are always considered as beneficial for organizations. Several researchers have presented their point of views on process improvement such as [48], [49], [50], [51]. The managers can better improve a process by avoiding false assumptions such as business improvement is dependent on process improvement, process change leads to process improvement, "software processes are non-lethal," and business process management activities do not need IT processes [52]. [53] has considered commitment as an important factor in software process improvement and has pointed out the four misconceptions in the existing commitment models for software process improvement. These and many other assumptions are considered as hindrances in process improvement and

management practices. Further, [52] has not found any direct relationship between software process change and performance. Process management and improvement processes become the direct responsibility of the project manager. In addition, lack of coordination among team members, and delayed reporting of problems and issues [54] in the project are also considered as one of the factors that affect the management process. In a research work on resource instantiation policies to automate software process management [55] has criticized on the existing process management tools and technologies that they “fall short in their computational capabilities and only provide passive project tracking and reporting aids”. The commitment to improve a software process is a basic integral part of the whole process since its beginning. Three forms of commitment such as affective, continuance and normative commitment are introduced by [53]. Many other researchers such as [56], [57], [58], [59], [60], [61], [62], [63] have discussed various other aspects of software process improvements. A prominent work in this regard is done by [56] in which a platform which is based on the CMMI, SPICE, PSP, and 6-Sigma standards is proposed to improve the capabilities of a software process. Emphasis is given to the continuous efforts of process improvement after its beginning. Project managers of small and medium size organizations consider CMMI and ISO standards of process improvement unnecessary [57]. However, [63] considers process standardization and process reuse as the same thing. Extensive documentation, limited resources, training cost, lack of guidance, unnecessary processes, practices and reviews are the limitations of CMMI to be adopted by small organizations [61]. Based on this argument, [63] proposed a meta-model to standardize the reusability of a software process integrating the people, roles, process and infrastructure components. The customer’s satisfaction, to the best of our knowledge, for the first time, was regarded as one of the factors in the software process improvement model by [57]. Software quality, cost, project scheduling and organization performance are the motivational factors for a project manager behind the innovative software processes [58], [62]. In addition to software process, skill level of software development team, tools and technologies, software complexity, deadlines, interaction and communication are also important factors that determine the software quality and organization performance [62].

Software development processes, models and frameworks are not the new areas for the researchers. Hence, IT globalization has given the new direction to it. As a consequence, new trends in software development processes have been emerged. Software process tailoring is one of the new emerging practices. Research works on software process tailoring is not available in quite a good number. In the next section we have discussed the prominent work on software process tailoring.

4 Software Process Tailoring – An Emerging Practice

The software process tailoring has emerged as a part of process improvement strategy. Though a number good quality research works on process tailoring have been presented by the researchers since 2000, but efforts had started back in 1980s. Software engineering researchers have realized the importance of this important practice and have produced some models and frameworks in recent years, but still a concrete effort is required from the software engineering research community.

In 1987, [64] presented a framework to tailor the software process as an evolutionary software improvement practice. In the framework process tailoring was performed based on the project goals and environments. The quantitative characterization through defect profile (analysis of errors, faults, and failures) classification scheme was made. The framework is supposed to be the beneficial contribution.

The process tailoring is of two types i.e. product tailoring and activity tailoring [65]. The tailoring approach formalized by [65] does not identify other tailoring activities except deletion and modification. The model provides very limited details on software tailoring process and is based on GV-model which is most likely not used by majority of the organizations. A software process is the key to success for a software development project. Software processes may be light weight or complex heavy weight approaches. Process tailoring is a standardized practice which is directly related with the tailoring of the activities of software development phases. During the process tailoring activity adoption and maintenance of a process standard and its reusability is necessary [66]. In the same work [66] has applied addition, deletion, splitting and merging process tailoring activities on a process module. The techniques such as correctness checking, conformance checking and compliance evaluation were used to verify the tailored process. This was considered as a good work on tailoring as four process tailoring activities were formally applied to a process. The need of a systematic approach for process tailoring and its verification was highlighted.

As mentioned earlier that IT globalization increased the use of agile based methodologies because of their fast paced development and light weight processes of software development. Due to this factor, most of the researchers focused on agile methodologies such as XP and scrum in their research works. In 2002, [67] applied their own tailored version of XP on a project and compared the differences, advantages and disadvantages with the standard XP approach. The small release plan, continuous integration of the components, use of pair programming, managing requirements only for the current build/milestone, planning and prioritizing by the developers, code refactoring, and testing were the practices adopted in their tailored version of XP. Except few issues, an improvement in the quality of their software was reported. In the same year [68] presented a framework to use process knowledge to tailor a software process and proposed a prototype tool to help to capture and use the process knowledge. The framework does not provide guidelines on types of information, and level and amount of information required to tailor a process. In a process tailoring framework [69] has defined two types of knowledge which are: 1) generalized knowledge that refers to the general rules, policies, standards and formulated information, 2) contextual knowledge that refers to the organizational decisions, events, time etc. [69] has beautifully expressed the effectiveness and efficiency of knowledge management in software process tailoring. In another work on knowledge based process tailoring [70] retrieved the most similar cases on the basis of existence of similar elements between past projects and a new project, and is applied to a new project. [70] retrieved the most similar cases (structural similarity)

through this approach and concluded that least modifications would be required in the process which would be generated through this approach. In 2004, [71] proposed a process tailoring framework that was also based on gathering information on existing techniques and practices and maintaining a knowledge base. Based on the knowledge repository activities such as problems analysis, finding the solution, proposed process implementation and its evaluation were recommended to perform. The model was more likely a similar kind of work as presented by [68], [69], [70].

The software process tailoring is directly related to the size of the organization and project's size, scale and scope. Software engineering research community is agreed that small and medium scale organizations prefer agile based methodologies for software development. Extreme programming (XP) is the most common approach being followed by majority of the companies. Therefore, varying sized companies and projects need to adapt XP or any other agile methodology according to their requirements. Authors in [72] used rule description practices (RDP) technique, similar to CRC cards, to tailor XP. Two type of rules were defined which are rules of engagement and rules of play. Rules of engagement were agile based and rules of play were based on XP. The RDP technique proved to be applicable in almost all types of environments. Other researchers such as [73], [74], [75], [76], [77], [78] as cited by [72] have presented very practical solutions on adopting XP according to different environments. [79], [80] in their research work claimed that it is a very common practice to partially adopt the XP.

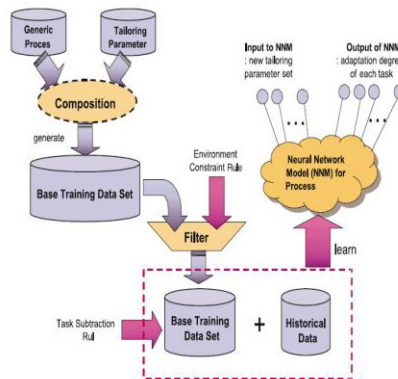


Fig. 2. Process Filtering Phase[84]

As a general procedure [81] performed tailoring of the software process meta-model [82]. Analysis of organizational environment, process life cycle, roles, activities and artifacts, documentation, training, and testing were the key elements of the procedure. The model did not provide any details of tailoring activities except presenting a general procedure to follow. The software engineering researchers have performed process tailoring from different aspects. The fundamental part of the process tailoring is the identification of reason or problem for which solution is provided through process tailoring. It deals with the analysis of various elements at

organizational level and project level. In a similar kind of work on process tailoring [83] have presented four types of strategies to analyze a system. The categorization is made on the basis of immediate needs and long term goals for each type. Based on the information achieved from the analysis of each strategy, the selection criteria are proposed. Similar to other works, the model provides another way of creating a knowledge base for process tailoring. A different approach for process tailoring using neural networks was presented by [84]. The process was accomplished through three phases namely process filtering, reconfiguration and feedback. During first phase tasks were selected from a generic process, then precedence was set during the reconfiguration phase and finally a tailored process was produced. During feedback phase the performance of the tailored process was evaluated. The filtering technique was the uniqueness of this approach and is shown in figure 2.

The authors have used more systematic but complex approach during the research.

The quality of a process as a result of process tailoring is very important for an organization. Bad process tailoring practices may affect the project budget, development time, quality of the software, compliance with the standards, and satisfaction of the employees. Also, the addition of unnecessary activities and omission of necessary activities are considered as wastage of time and money [85]. As a result of review of literature available on the software process tailoring [85] has reported 1) focusing on project level or organizational level, 2) case study in real environment, 3) size of the company, and 4) supporting tools etc as the major issues in process tailoring. The authors in this study have presented the review of existing software process tailoring approaches from almost all aspects, thus providing a guideline for future purposes. In a research work, [86] has performed four process tailoring operations namely addition, deletion, splitting and merging based on structured Petri Net without initial marks denoted as basic block. All four operations were based on four basic blocks namely, sequence block, selection block, iteration block and concurrency block. The approach followed in the study is more systematic as compared to other works but is unable to answer the questions raised by [85]. For project level process tailoring, a four step iterative approach was presented by [87]. Evaluation of the project's goals and environment, assessment of the challenges faced by projects, finding the suitable process tailoring strategy and lastly validation and evaluation of the tailored software process operations were performed to tailor a process. The authors have proposed an iterative approach to make it more flexible because they believe that processes gets more refined with the progress of the project. On contrary, they are also not in the favor of excessive and repeating process tailoring during a project lifecycle.

Software process tailoring is a part of process improvement approaches. All the organizations irrespective of their sizes adapt a process model according to their needs. Size of the project and organization, project scale, its complexity vary from one project and organization to other project and organization. Even a process model suitable for a project, might not be beneficial for the other project in the same organization.

Software process tailoring is an emerging trend because of the un-decidability factor involved in the selection of a suitable process according to the requirement of the project and organization. During our analysis we faced the limitation of the availability of literature on process tailoring. Though work on process tailoring had

started in 1980s but software engineering research community could not pay deserved attention to this practice. Now a days majority of the software development companies are relying on software process tailoring approaches and it has found a prominent place in software engineering research works.

5 Conclusion

Process improvement practices are the continuous part of project management and standardization processes. The work presented by the researchers in the areas of agile, process improvement and tailoring is very general. It neither addresses the industry issues nor meets their requirements. Most of the work has poor methodology and is not properly validated. Results do not show the applicability of these models and frameworks. Further, it is found that among all approaches, the process tailoring is the most smart and efficient practice to solve the process and projects related issues. It has emerged as a lightweight and faster approach. Almost all types of companies adopt process tailoring techniques. In support of agile methodologies, they have become the new generation of processes. As a newly emerging trend, software process tailoring needs more contribution from the researchers and practitioners. There is a need of more formal and applied approaches, methodologies, frameworks and standards in this regards. The involvement of the actual practitioners in the research process is required. In its current form, it is unable to provide solutions to the IT industry. More concrete efforts both by the software engineering researchers and actual practitioners are required through joint projects. During the coming years, process tailoring would be the ultimate choice of the companies. It has been realized that smart processes are being preferred by the industry and process tailoring is just the beginning.

References

1. Ramasubbu, N., Ballan, R.K.: Globally Distributed Software Development Project Performance: An Empirical Analysis. In: ESEC-FSE 2007, Cavtat near Dubrovnik, Croatia, September 3–7, pp. 125–134 (2007)
2. Cho, J.: Globalization and Global Software Development, Issues in Information Systems, vol. VIII (2), pp. 287–290 (2007)
3. Ktata, O., Levesque, G.: Agile development: Issues and Avenues Requiring a Substantial Enhancement of the Business Perspective in Large Projects. In: C3S2E 2009, Montreal Qc, Canada, May 19-21, pp. 59–66 (2009)
4. Akbar, R., Hassan, M.F., Safdar, S., Qureshi, M.A.: Client's Perspective: Realization as a New Generation Process for Software Project Development and Management. In: ICCSN 2010, pp. 191–195 (2010)
5. Akbar, R., Hassan, M.F.: A Collaborative-Interaction Model of Software Project Development: An Extension to Agile Based Methodologies. In: ITSIM 2010, pp. 1–6 (2010)
6. Rao, N.M.: Challenges in Execution of Outsourcing Contracts. In: ACM ISEC (2009)
7. Sterba, C., Grechenig, T., Pazderka, M.: Outsourcing as a Strategy for IT Harmonization – A Public Sector Case Study Proposing an Approach in Independent Stakeholder Scenarios. In: ICEGOV 2008, Cairo, Egypt, pp. 245–250 (2008)
8. Taylor, H.: The Move to Outsourced IT Projects: Key Risks From the Provider Perspective. In: SIGMIS-CPR 2005, pp. 149–154 (2005)

9. Kolawa, A.: Out Sourcing Devising a Game Plan. Queue (2004)
10. Gopal, A., Mukhopadhyay, T., Krishnan, M.S.: The Role of Software Processes and Communication in Offshore Software Development. Communications of the ACM 45(4ve), 193–200 (2002)
11. Narayanaswamy, R., Henry, R.M.: Effects of Culture on Control Mechanisms in Offshore Outsourced IT Projects. In: SIGMIS-CPR 2005, Atlanta, Georgia, USA, pp. 139–145 (2005)
12. Ramasubbu, N., Balan, R.K.: Towards Governance Schemes for Distributed Software Development Projects. In: SDG 2008, Leipzig, Germany. pp. 11-14 (2008)
13. Aoyama, M.: Web-Based Agile Software Development. IEEE Software 15(6), 56–65 (1998)
14. Cusumano, M.A., Yoffie, D.B.: Software Development on Internet Time. Computer IEEE 32(10), 60–69 (1999)
15. Turk, D., France, R., Rumpel, B.: Limitations of Agile Software Processes. In: Proceedings of 3rd International Conference on eXtreme Programming and Agile Processes in Software Engineering. ACM, New York (2002)
16. Theunissen, W.H.M., Kourie, D.G., Watson, B.W.: Standards and Agile Software Development. In: Proceedings of SAICSIT 2003, pp. 1–11 (2003)
17. Nerur, S., Mahapatra, R., Mangalaraj, G.: Challenges of Migrating to Agile Methodologies. Communications of the ACM 48(5), 73–78 (2005)
18. Beck, K., Fowler, M.: Planning Extreme Programming. Addison-Wesley, Reading (2000)
19. Fraser, S., Beck, K., Cunningham, W., Crocker, R., Fowler, M., Rising, L., Williams, L.: Hacker or Hero? - Extreme Programming Today. In: Proceedings of the 2000 ACM Conference on Object-Oriented Programming, Systems, Languages and Applications (OOPSLA 2000), Minneapolis, MN, USA, pp. 5–7 (2000)
20. Schwaber, K., Beedle, M.: Agile Software Development with Scrum, 1st edn. Prentice Hall, New Jersey (2001)
21. Rising, L., Janoff, N.S.: The Scrum Software Development Process for Small Teams. IEEE Software, 2–8 (2000)
22. Glass, R.L.: Agile versus Traditional: Make Love Not War, Cutter IT Journal 14(12), 12–18 (2001)
23. Turner, R., Jain, A.: Agile Meets CMMI: Culture Clash or Common Cause? In: Wells, D., Williams, L. (eds.) XP 2002. LNCS, vol. 2418, pp. 153–165. Springer, Heidelberg (2002)
24. Fritzsche, M., Keil, P.: Agile Methods and CMMI: Compatibility or Conflict. e-Infomatica Software Engineering Journal 1(1), 9–26 (2007)
25. Paetsch, F., Eberlein, A., Maurer, F.: Requirements Engineering and Agile Software Development. In: Proceedings of the Twelfth International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, p. 308 (2003)
26. Eberlein, A.: Requirements Engineering and Agile Methods: Can they benefit from each other? In: Position Statement in the Proceedings of Canadian Invited Workshop on Scaling XP/AgileMethods, Banff, Canada (2003)
27. Basili, V.R.: The Role of Experimentation in Software Engineering: Past, Current, and Future. In: Proceedings of ICSE, vol. 18, pp. 442–449 (1996)
28. Jiang, L., Eberlein, A.: Towards A Framework for Understanding the Relationship between Classical Software Engineering and Agile Methodologies. In: APSO 2008, Leipzig, Germany, pp. 9–14. ACM, New York (2008)
29. Tarawneh, H., Elsheikh, A., Lahawiah, S.: Web-Based Applications Development in Small Firms. In: Proceedings of the 6th WSEAS Int. Conf. on Software Engineering, Parallel and Distributed Systems, Corfu Island, Greece, February 16-19, pp. 69–74 (2007)

30. Aiken, J.: Technical and Human Perspective on Pair Programming. *ACM SIGSOFT Software Engineering Notes* 29(5), 1–14 (2004)
31. Wills, G.B., Abbas, N., Chandrasekharan, R., Crowder, R.M., Gilbert, L., Howard, Y.M., Millard, D.E., Wong, S.C., Walters, R.J.: An Agile Hypertext Design Methodology. In: HT 2007, Manchester, England, UK, September 10–12, pp. 181–184. ACM, New York (2007) ISBN: 978-1-59593-820-6/07/0009
32. Souza, V.E.S., Felbo, R.d.A.: An Agile Approach for Web Systems Engineering. In: *WebMedia 2005*, Poços de Caldas, Minas Gerais, Brazil, December 5-7, pp. 1–3 (2005)
33. Ferreira, C., Cohen, J.: Agile Systems Development and Stakeholder Satisfaction: A South African Empirical Study. In: *Proceedings of SAICSIT 2008*, October 6-8, pp. 48–55 (2008)
34. Fernandes, J.M., Duarte, F.J.: A Reference Framework for Process-Oriented Software Development Organizations. *Software and Systems Modeling* 4(1), 94–105 (2005), doi:10.1007/s10270-004-0063-0
35. Krishnan, M.S., Mukhopadhyay, T., Zubrow, D.: Software Process Models and Project Performance. *Information Systems Frontiers* 1(3), 267–277 (1999)
36. Card, D.N.: Research Directions in Software Process Improvement. In: *Proceedings of the 28th Annual International Computer Software and Applications Conference, COMPSAC 2004* (2004)
37. Akbar, R., Hassan, M.F., Abdullah, A., Safdar, S., Qureshi, M.A.: An Insight into Real Software Industry Paradigms and Software Engineering Research. In: *International Symposium on Computers & Informatics* (March 2011)
38. Xia, F.: What's Wrong with Software Engineering Research Methodology. *ACM SIGSOFT Software Engineering Notes* 23(1), 62–64 (1998)
39. Luckowicz, P., Heinz, E.A., Prechelt, L., Tichy, W.F.: Experimental Evaluation in Computer Science: A Quantitative Study. *Journal of Systems and Software* 28(1), 9–18 (1995)
40. Tichy, W.F.: Should Computer Scientist Experiment More? 16 Excuses to Avoid Experimentation. *IEEE Computer* 31(5) (1997)
41. Zelkowitz, M.V., Wallace, D.: Experimental Validation in Software Engineering. *Information and Software Technology* 39(11), 735–744 (1997)
42. Zelkowitz, M.V., Wallace, D.R.: Experimental Model for Validating Technology. *IEEE Computer* 31(5), 23–31 (1998)
43. Shaw, M.: What Makes Good Research in Software Engineering? *International Journal of Software Tools for Technology Transfer* 4(1), 1–7 (2002)
44. Marcos, E.: Software Engineering Research versus Software Development. *ACM SIGSOFT Software Engineering Notes* 30(4) (2005)
45. Seaman, C.B.: Qualitative Methods in Empirical Studies of Software Engineering. *IEEE Transactions on Software Engineering* 25(4), 557–572 (1999)
46. Runeson, P., Host, M.: Guideline for Conducting and Reporting Case Study Research in Software Engineering. *Empirical Software Engineering* 14(14), 131–164 (2009)
47. Bahrami, A.: Integrated Process Management: From Planning to Work Execution. In: *BSN 2005 Proceedings of the IEEE EEE 2005 International Workshop on Business Services Networks*, pp. 11–11 (2005)
48. Hansen, B., Rose, J., Tjornehoj, G.: Prescription, Description, Reflection: the Shape of the Software Process Improvement. *International Journal of Information Management* 24(6), 457–472 (2004)
49. Herbsleb, J.D., Goldenson, D.R.: A Systematic Survey of CMM Experience and Results. In: *Proceedings of the 18th International Conference on Software Engineering*, pp. 323–330 (1996)
50. Stelzer, D., Mellis, W.: Success Factors of Organizational Change in Software Process Improvement. *Software Process - Improvement and Practice* 4(4), 227–250 (1999)

51. Rainer, A., Hall, T.: Key Success Factors for Implementing Software Process Improvement: A maturity-Based Analysis. *Journal of Systems and Software* 62(2), 71–84 (2002)
52. Bannerman, P.L.: Capturing Business Benefits from Process Improvement: Four Fallacies and What to Do About Them. In: *BIPI 2008*, Leipzig, Germany, May 13, pp. 1–8 (2008)
53. Abrahamsson, P.: Commitment Development in Software Process Improvement: Critical Misconceptions. In: *23rd International Conference on Software Engineering, ICSE 2001*, pp. 71–80 (2001)
54. Keil, M., Smith, H.J., Pawlowski, S., Jin, L.: Why Didn't Somebody Tell Me?: Climate, Information Asymmetry, and Bad News About Troubled Projects. *ACM SIGMIS* 35(2), 65–84 (2004)
55. Reis, C.A.L., Reis, R.Q., Schlebbe, H., Nunes, D.J.: A Policy-Based Resource Instantiation Mechanism to Automate Software Process Management. In: *SEKE 2002*, Ischia, Italy, July 15–19, pp. 795–802. ACM, New York (2002) ISBN: 1-58113-556-4/02/0700\$5.00
56. Kim, J.A., Choi, S.Y., Kim, T.H.: Management Environment for Software Process Improvement. In: *International Symposium on Computer Science and its Applications*. IEEE, NY (2008)
57. Xiaoguang, Y., Xiaogang, W., Linpin, L., Zhuoning, C.: Research on Organizational-level Software Process Improvement Model and Its Implementation. In: *International Symposium on Computer Science and Computational Technology*, pp. 285–289 (2008)
58. Guo, Y., Seaman, C.B.: A Survey of Software Project Managers on Software Process Change. In: *ESEM 2008*, pp. 263–269 (2008)
59. Morgan, B., Lowry, G.: Software Process Assessment and Improvement, Discussion Summary. *IEEE Explore*, 497–499 (1996)
60. Bueno, P.M.S., Crespo, A.N., Jino, M.: Analysis of an artifact oriented test process model and of testing aspects of CMMI. In: Münch, J., Vierimaa, M. (eds.) *PROFES 2006*. LNCS, vol. 4034, pp. 263–277. Springer, Heidelberg (2006)
61. Brodman, J.G., Johnson, D.L.: A Software Process Improvement Approach Tailored for Small Organizations and Small Projects. In: *Proceedings of International Conference on Software Engineering (ICSE 1997)*, pp. 661–662 (1997)
62. Paulish, D.J., Carleton, A.D.: Case Studies of Software Process Improvement Measurement. *IEEE Computer* 27(9), 50–57 (1994)
63. Succi, G., Benedicenti, L., Predonzani, P., Vernazza, T.: Standardizing the Reuse of Software Processes. *Standard View - Supporting Article* 5(2), 74–82 (1997)
64. Basili, V.R., Rombach, H.D.: Tailoring The Software Process To Project Goals and Environments. In: *ICSE 1987*. ACM, New York (1987)
65. Welzel, D., Hausen, H.-L., Schmidt, W.: Tailoring and Conformance Testing of Software Processes: the ProcePT Approach. In: *Proceedings of the 2nd IEEE Software Engineering Standards Symposium*, pp. 41–49 (1995)
66. Yoon, I. C., Min, S.Y., Bae, D.H.: Tailoring and Verifying Software Process. In: *APSEC 200*. *IEEE Explore* (2001)
67. Bowers, J., May, J., Melander, E., Baarman, M., Ayoob, A.: Tailoring XP for Large System Mission Critical Software Development. In: Wells, D., Williams, L. (eds.) *XP 2002*. LNCS, vol. 2418, pp. 269–301. Springer, Heidelberg (2002)
68. Xu, P., Ramesh, B.: A Tool for the Capture and Use of Process knowledge in Process Tailoring. In: *Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS 2003)*, IEEE Computer Society, Los Alamitos (2002), 0-7695-1874-5/03 \$17.00 ©
69. Xu, P.: Knowledge Support in Software Process Tailoring. In: *Proceedings of the 38th Hawaii International Conference on System Sciences* (2005)

70. Kang, D., Song, I.G., Park, S., Bae, D.H., Kim, H.K., Lee, N.: A Case Retrieval Method for Knowledge-Based Software Process Tailoring Using Structural Similarity. In: 15th Asia-Pacific Software Engineering Conference. IEEE, Los Alamitos (2008) doi: 1530-1362/08 \$25.00 ©
71. Keenan, F.: Agile Process Tailoring and problem anALYsis (APTLY). In: Proceedings of the 26th International Conference on Software Engineering (ICSE 2004), pp. 45–47 (2004), 0270-5257/04 \$20.00 © 2004 IEEE
72. Mirakhorli, M., Rad, A.K., Shams, F., Pazoki, M., Mirakhorli, A.: RDP Technique: a Practice to Customize XP. In: APOS 2008, Leipzig, Germany, pp. 23–32 (2008)
73. Murru, O., Deias, R., Mugheddo, G.: Assessing XP at a European Internet Company. *IEEE Software* 20(3), 37–43 (2003)
74. Drobka, J., Noftz, D., Raghu, R.: Piloting XP on Four Mission-Critical Projects. *IEEE Software* 21(6), 70–75 (2004)
75. Grenning, J.: Launching Extreme Programming at a Process Intensive Company. *IEEE Software* 18(6), 27–33 (2001)
76. Cao, L., Mohan, K., Xu, P., Ramesh, B.: How Extreme Does Extreme Programming Have to Be? Adapting XP Practices to Large-Scale Projects. In: Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS 2004) - Track 3, vol. 3 (2004)
77. Elssamadisy, A.: XP on a large project - A Developer's view. In: Proceedings of XP/Agile Universe, Raleigh, NC (2001)
78. Lui, K.M., Chan, K.C.C.: A road map for implementing eXtreme programming. In: Li, M., Boehm, B., Osterweil, L.J. (eds.) *SPW 2005*. LNCS, vol. 3840, pp. 474–481. Springer, Heidelberg (2006)
79. Reifer, D.J.: How to get the most out of extreme programming/Agile methods. In: Wells, D., Williams, L. (eds.) *XP 2002*. LNCS, vol. 2418, pp. 185–196. Springer, Heidelberg (2002)
80. Aveling, B.: XP lite considered harmful? In: Eckstein, J., Baumeister, H. (eds.) *XP 2004*. LNCS, vol. 3092, pp. 94–103. Springer, Heidelberg (2004)
81. Ibraguengoitia, G., Salazar, J.A., Sanchez, M.G., Ramirez, A.Y.: A Procedure for Customizing a Software Process. In: Proceedings of the Fourth Mexican International Conference on Computer Science (ENC 2003), p. 68 (2003)
82. Oktaba, H., Gonzalez, G.I.: Software Process Modeled with Objects: Static View. *Computacion y Sistemas* 1(4), 228–238 (1998)
83. Bustard, D.W., Keenan, F.: Strategies for systems analysis: groundwork for process tailoring. In: *ECBS 2005*, pp. 357–362 (2005)
84. Park, S., Na, H., Park, S., Sugumaran, V.: A Semi-Automated Filtering Technique for Software Process Tailoring Using Neural Network. *Expert Systems with Applications* 30(2), 179–189 (2006)
85. Pedreira, O., Piattini, M., Luaces, M.R., Brisaboa, N.R.: A Systematic Review of Software Process Tailoring. *ACM SIGSOFT Software Engineering Notes* 32(3) (2007)
86. Dai, F., Li, T.: Tailoring Software Evolution Process. In: Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, pp. 782–787. IEEE Computer Society, Los Alamitos (2007), doi:10.1109/SNPD.2007.25, 0-7695-2909-7/07 \$25.00 ©
87. Xu, P., Ramesh, B.: Using Process Tailoring to Manage Software Development Challenges. *IEEE Computer Society ITPro* 10(4), 39–45 (2008)

An Evaluation Model for Software Reuse Processes

Anas Bassam AL-Badareen, Mohd Hasan Selamat, Marzanah A. Jabar,
Jamilah Din, and Sherzod Turaev

Faculty of Computer Science and Information Technology
University Putra Malaysia

Anas_badareen@hotmail.com,
{hasan,marzanah,jamilah,sherzod}@fsktm.upm.edu.my

Abstract. Software reuse is a major concern in many software development companies. It is one of the main strategies used to reduce the cost of software product development. Studies show that the reuse strategy is the most significant strategy in terms of effort and quality. That it could save the half of the software development effort and increase the quality of the software product. Different ways of software reuse are proposed and discussed. In this study, an evaluation model for software reuse is proposed. The model is developed in order to consider the new methods of software reuse. That developed based on the framework of develop a reusable software components through software development processes. The model is proposed in order to present the applicable methods of software reuse and to evaluate their cost.

Keywords: Reusability, Software Reuse, Reuse Strategy, Software Development, Reuse Framework, Reuse Scenario, develop for Reuse, Develop by Reuse.

1 Introduction

Software reuses is a process of design and develops software assets, and then reuse these assets to develop other software products in the future. It consist of two main tasks, develop-for-reuse and develop-by-reuse [1]. Develop-for-reuse is a process of produce software assets that able to be adapted in different software products [2-3]. Develop-by-reuse is a process of adapt and includes an existing software asset in order to develop a new software product [4]. This strategy is used to reduce the time and expenses of developing and enhance the flexibility, maintainability, and reliability of software product [5-8]. However, the reuse is not limited to specific stage or components; it can be occurred at all stages of the development in different forms, from code, product components, designs and architectures, to skill and knowledge [9].

According to Boehm [10] software reuse is one of three main strategies are used to increase the productivity of the software development companies. It is used to avoid developing unnecessary work by reusing software artifacts instead of custom developing each project. This strategy approximately saves half of the effort required

in normal software development process. Therefore, the reuse concept is rapidly used in software industry and considered as a one of the main research interest.

Ramachandran [11] develop a method of develop, asses, and reuse a reusable software assets. The method consists of development guideline and automated tools that can provide an advice and analysis in develop by reuse process.

Ravichandran & Rothenberger [12] defined three main reuse strategies, Black-Box Reuse (CBD) with component markets, Black-Box Reuse (CBD) with internal components, and White-Box Reuse with internal components. An extensive comparison has been conducted between those strategies. Moreover, a decision tree of component reuse is developed.

Tomer [13] developed an evaluating model for software reuse and presented four applicable scenarios for software reuse. A comparison between those scenarios and the normal development was conducted based on seven industrial software assets. In this study, we intend to use the model proposed in [13] and the reuse framework [1] in order to presents the applicable ways of develop and reuse a software asset.

In this study, we intend to discuss the framework of reuse software components, analyze the current model of software reuse evaluation, and propose a new model of software reuse evaluation. Section two, discussed the framework of software reuse, section three discuss the proposed model of software reuse evaluation, section four, presents the applicable scenarios of software reuse in both sides, develop for reuse and develop by reuse. The discussion and implication of the proposed model in section five, and section six conclude the study and presents the intended work in the future.

2 Reusable Software Component Framework

Reusable software component framework proposed in [1-2]. The framework intends to develop and extract software components during software product development in order to be stored and reused in other software product. This allows using the effort of developing a software product efficiently and enhances the method of software reuse. The proposed framework considers both develop a normal software asset and develop a reusable software asset during software product development.

Basically, the main goal of software product development is to develop software components that able to achieve the defined requirements for that product, whereas, the framework of software reuse intends to add an extra goal. This goal is to consider the developed assets in order to be used in the future in other software products (*See figure 1*).

In order to achieve the reusability objectives, two main ways are applicable, develop a normal component and develop a reusable component. The normal component is the traditional way of developing a software product, which it aims to develop a software component in order to achieve the defined requirements. In this process, the component is extracted from the product during development as it is and cataloged in the repository. The second way is to develop a reusable software component instead of normal asset, and then extract and cataloged the component in the repository. In this way, an extra effort is added to the development process in order to achieve the reusability requirements.

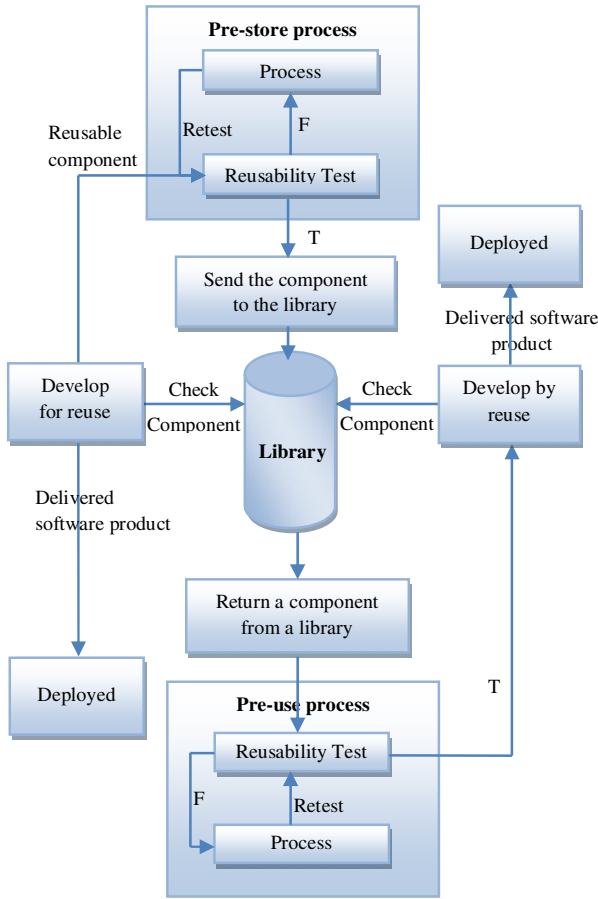


Fig. 1. Reusable software component life cycle

3 The Evaluation Model

The reusable asset could be in one of two main forms, private asset and repository asset [13]. Private asset is a software component included in specific software product, which it is collected from the repository, other products, or developed from scratch. This asset is available for mining and modification in order to be stored or reused in other software product. Repository asset is a software component cataloged and stored in reusable software repository. It is developed and used in several ways on the contrary of the normal software assets, that it is created during software development and used only on the intended software product.

The reuse strategy consists of two main tasks, develop-for-reuse and develop-by-reuse. Develop-for-reuse is the process of produce a software components that might be used in other software products in the future. Develop-by-reuse is the process of

adopt and used an existing software products or it is parts in order to produce a new software product.

Whereas, in different ways both of develop-for-reuse and develop-by-reuse tasks can be conducted. Tomer [13] proposed a model of software reuse evaluation model. The model presents the applicable ways of produce and reuse software components. It shows the applicable operations and transitions that can be performed on the components in software reuse process. These operations and transitions were defined based on the two of three dimensions evaluation of software product line model proposed in [14], Transition operations and transformation operations. However, the new method of software reuse is not considered in this model. Therefore, a new structural model for software reuse evaluation is developed (*See figure 2*).

The model considered all applicable ways of software reuse from both develop-for-reuse and develop-by-reuse. A new type of develop for reuse is considered. Therefore, software products are classified in three main categories, existed software product, product during development, and product under development. Moreover, the assets are classified into three main categories, normal asset, reusable asset with internal components, and reusable asset with market components.

3.1 Software Products

- **Existed software product:** is the software has been developed previously in house and is available for mining it is components in order to be cataloged in the repository or reused in new software products. Whereas, the reusability concept is not considered in this product, the components exist in this software are only a normal software assets. Moreover, the assets existed in the system are not categorized in the repository. Therefore, the developers have to know about the existing software products and an extra effort for asset mining is required.
- **Product during Development:** is an existing software product, but the reusability has been defined early during software product development and the components are cataloged in the repository directly. In this type of development, the probability of know and remember an existing software product is dropped. Moreover, the reusability can be considered in two ways.
 - Develop a reusable asset instead of normal asset, which will add an extra cost to the development, but it will reduce the cost of develop by reuse process and increase the probability of reuse the asset.
 - Develop a normal asset, which will be cataloged directly in the repository. In this process, the cost of mining the asset later on from the existing product is avoided.
- **Product under Development:** is a software product currently is under development process or managed to be developed. This product is managed to be reuse the existing software components from the repository, exist software product, or develop a new software components.

The relationships are defined in this model based on the elementary operations were defined in [13], whereas new operations are defined, existed operation are divided into two or more tasks, and others are used as it is. The main differences with the previous model, the operation are presented in a structural way. Therefore, the

external relationships that link the products with each other and with the repository presented the transition and the relationships between the assets within the products or the repository present the transformation operation.

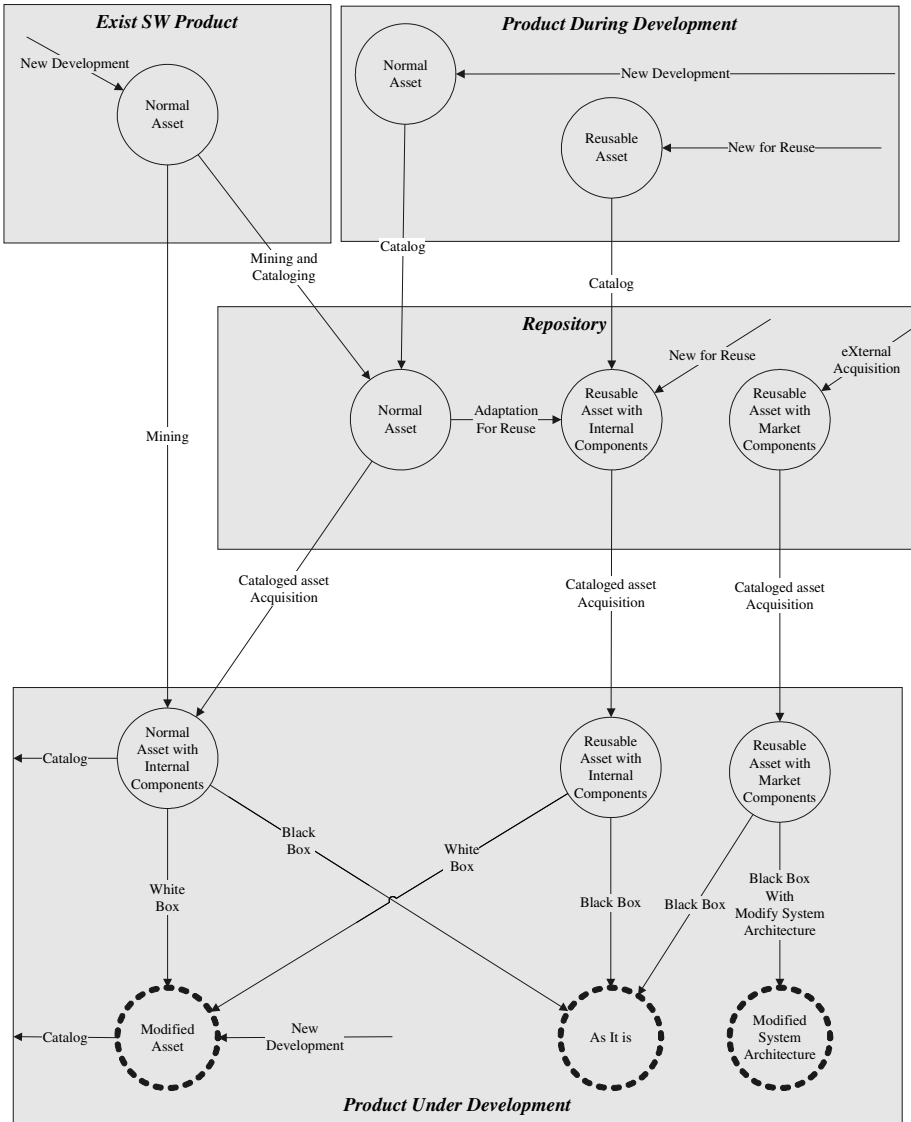


Fig. 2. Elementary Reuse, Operations and Transitions

3.2 Assets Repository

The assets repository is the library of the software assets that developed and imported from different resources in different ways. The repository stores the different versions of the reusable asset, normal assets, reusable assets with internal components, and reusable assets with market components. The importance of the repository is that categorized and classified the reusable assets in a certain way in order to simplify and enhance the process of categorizing, mining, retrieving, and reusing the reusable assets. That will reduce the cost of reuse process and increase the probability of using the existing assets as many as can.

In order to increase the probability of using specific assets and enhance them to be applicable for different functions and platform, the adaption for reuse process (AR) is considered in the repository side. This process is intended to enhance the exiting assets in the repository to increase the chance of using them in different systems as many as can. Moreover, it is used to increase the probability of using the assets as a black box instead of white box reuse, which will save the extra efforts that required in white box reuse.

3.3 Assets Used in Software Reuse

Tomer [13] defined two types of software assets based on the state of the asset, whether it is stored in the repository or existed in certain software product. In this model we defined the different types of the assets that are included in the process. We defined them based on the state of the asset in terms of characteristics, in addition to the location. Moreover, the different operations and transition are considered and calculated.

Normal Asset (NA): is the asset developed for specific function in certain software product. Normally, this type of asset is conducted during software product development. Therefore, it contains internal software components and allowed to be modified. This asset can be used directly in other software product with or without modification or it can be cataloged first in the repository as it is, whereas an adaptation for reuse is required to be able to be used in different types of software products and it will be defined as a reusable component.

Reusable Asset with Internal Components (RA_{IC}): is a reusable component contains the internal source and allows a modification in order to fit new environments and conditions. Normally, this type of assets is in house development, which it can be developed during software product development, developed independently and stored in the repository, or resulted from modifying a normal assets cataloged in the repository.

Reusable Asset with Market Components (RA_{MC}): is a reusable asset imported from external sources. Therefore, it contains a market component that does not allow any modification. This component it can be used only as a black box, while any modification required will be conducted on the system components to fit the new component.

3.4 Reuse Operations

The process of software reuse consists of different operations that required performing a reuse task. As we defined above, two main operations are applicable in software reuse, transition operation and transformation operations.

The **transition operation** is a process of transfer a software asset between software products or between software products and software repository. This process is classified into the following tasks:

- *Catalog (C)*: the process of catalog and store the imported software component in the repository. The asset is imported from an existing software product or from software product during development.
- *Mining (M)*: is the process of search and defined a software asset within existed software product. The process of mining software asset is conducted on an existed software product that is the developer knows that the intended system contains a required asset.
- *Cataloged asset acquisition (CA)*: the process of mining and defined a software asset within a repository. The developers search for a required software asset in the repository in order to be used in the new software product.
- *External Acquisition (XA)*: the process of import a software asset from external sources and catalogs it in the repository. This process is similar to the process of catalog an asset, whereas usually this operation is conducted on reusable assets with market components (out sources development). While the cataloged operation is used for in house development assets to store a normal assets or a reusable assets with internal components.

The **transformation operation** is the process of modify software asset and change its characteristics from one state to other. This process is conducted within software product process or with software reuse repository.

- *New Development (ND)*: the process of develop a software asset from scratch.
- *New for Reuse (NR)*: the process of develop an internal reusable asset from scratch (Reusable Asset with Internal Components).
- *Adaptation for Reuse (AR)*: the process of modify an existed software asset in order to be reuse and suitable for other software products. Normally, this type of operation is conducted on the normal assets were imported from previous software products, either existed software products or during software product development.
- *Black Box with Modify System Architecture (BB_{MSA})*: is the process of modify system architecture in order to adapt a new reusable asset. This task is defined as a side effect of external reusable software that is this type of assets does not allow any modification, only the configuration through defined parameters is possible [12]. Therefore, a software system has to be suitable to adapt this asset, which need a modification in its architecture.

- *Black Box as it is (BB_{AI})*: is a process of adapt a reusable asset in the new system without any modification neither the asset nor the system architecture. At this state, the software asset is fixed and achieves all of the requirements that needed in the new system.
- *White Box (WB)*: is the process of modify an existed software asset either normal or reusable assets in order to be adapted in the new system. Normally, this process is conducted on the normal assets were in house developed. While, the reusable assets with the internal components allows this process but it is not common. That is the reusable assets considered the main requirements for new software and any modification required is a related to the adaptation requirements and parameters configuration.

4 Software Reuse Scenarios

In different ways software assets are developed and reused. This section presents the applicable scenarios of develop, catalog, acquisition, and reuse software assets. Though, the process of reuse software assets is based on the state of the defined asset, not on how the asset is developed. Therefore, we divided the scenarios into two parts based on the sides of the reuse process, develop for reuse and develop by reuse. In the following, the applicable scenarios of develop and reuse software assets.

4.1 Develop-for-Reuse (DFR)

The develop-for-reuse is the produce, extract, or import software components intended to be used in the future. These types of assets can be collected and produces in different ways as defined below.

Extract from existing software product (EE): the required asset is included in specific existing software product. This process consists of mining the asset in software product and cataloged it in the repository. The extracted asset does not satisfy the reusability characteristics it is only depends on the developers' knowledge about the existing software products.

New for Reuse (NR): the reusable asset is developed from scratch in order to be used in different software products. The reusability characteristics are considered during development, such as generality, interoperability, and co-existence. The cost of this type of software component is higher than that required to develop a normal. That is the reusability characteristics are considered during development. However, this type of software components rarely required a modification in order to be adopted in software products. The modification required is only to be fit in the architecture of the new software product.

External acquisition (XA): the asset is out source software component acquired from some external resources (COTS) artifact. This asset same as new for reuse asset figure on the reusability characteristics, whereas it can be used only as a black box reuse without source code notation. The advantage of this component is that only used to reduce the time of software development. However, while this type of asset does not allow a modification, it required to modify system architecture in order to be fit in the system.

Tomer [13] considered the cost of import external acquisition as the cost of cataloged the asset in the repository, while the marketing price is excluded. We estimate the cost of import external asset (COTS) is equal to the cost of developing a new reusable asset. The difference only the cost of cataloged the external asset in the library, whereas the internal asset does not require this task.

Adaptation for Reuse (AR) is the process of modifying repository asset in order to develop other repository asset. In this case, any modification occurred on the repository assets in order to achieve certain goals or satisfy specific adopting condition is cataloged in the repository as a new version of the existing asset. However, this modification normally is conducted on the normal asset cataloged in the repository. The cost of this process includes the mining, modifying, and cataloged the new version in the repository.

Extract during Development [6]: the intended software component is defined early in software development in order to be extracted and cataloged in the repository. In this method, two cases are applicable, develop a normal component or develop a reusable component.

Extract Reusable Asset during Development (RAD): at the same time of developing a certain asset for specific software product, this asset is developed as a reusable asset. This way of asset developing is similar to the developing a new asset for reuse, whereas in this case the asset is a part of developing a specific software product in contract with the developing a reusable asset that it is developed independently. Therefore, an extra cost is paid, this extra cost is the distinction between develop a new asset and develop a new reusable asset.

Extract Normal Asset during Development (NAD): during developing specific software product, the asset needed to be cataloged in the repository is considered. This way of asset extraction is same as extract from existing software product in terms of asset characteristics, whereas the differences in the cost mining and the probability of finding the asset. The cost of extract the asset during developing specific software product is only the cost of cataloged the asset in the repository and no any asset mining required in this process. Extract the asset from existing software product is based on the developer knowledge about the existing software product, whereas in the extraction during development does not require any knowledge about existing systems or remember any previous assets.

4.2 Develop-by-Reuse (DBR)

Develop by reuse is a process of retrieve an existing software components either normal or reusable components in order to be used in new software product. Whereas, three types of software components are used in this process, normal asset, reusable asset with internal components, and reusable assets with market components. Therefore, the develop-by-reuse process is conducted based on the type of existing assets. Two main types of develop-by-reuse are defined based on the modification required on the software component, black box reuse and white box reuse.

Black Box reuse (BB) is the process of reuse software asset as it is, whereas both of internal and external assets are used. In the internal assets, if the asset completely

satisfied the new system requirement and there is no any modification required, it will be considered as a black box. Otherwise it will be considered as a white box.

In the external asset (COTS), black box is the only way it can be used to reuse this type of asset. That is the developers are not allowed to modify it. Therefore, the cost of reuse this asset is that required to retrieve and adopt it in the new system.

However, the problem of this type of asset is that any modification required in order to adopt it in the new system will be conducted on the system components instead modify of the asset. This will increase the cost of reuse the asset instead of the main goal which is decreasing it. Therefore, if any modification required in the system in order to adapt the new asset (COTS) will include the cost of modify the system components.

White Box reuse (WB) is the process of modify existing software asset in order to be reused in specific software product. In this case, only the internal assets are used, whereas the external assets (COTS) are not able to be. That is the external assets does not offered the internal contents of the software components, it is only offered information about the external characteristics.

The internal asset that is developed in house and it is able to be used as a black box and white box. The external asset is an outsource components (COTS) and it is able to be used only as a black box. In the internal assets, if it is satisfied the new system requirements and adoption it will be used without any modification, where if any modification is required to achieve a system requirements or adoption it will be considered as a white-box reuse.

The external asset (COTS) is that the external sources software components and it is developed for specific goals with a certain conditions. It is used only as a black box without any modification allowed. Therefore, it must satisfy the system requirements and for adoption the system components are required to be modified in order to import this asset. In this case, the external asset required more effort to modify the system components in order to be adopted.

5 Discussion and Implication

In this study, the process of software reuse is discussed from two sides, develop for reuse and develop by reuse. The study proposed a new model of software reuse evaluation. The model considered the new method of software reuse and presents the different types of software components based on their characteristics. Three main types of software components are considered and classified in this study, normal asset, reusable asset with internal components, and reusable asset with market components.

The normal asset is developed during software product development process, this type of components always modified in order to be used in other products. The cost of this asset in the reuse process is that only required for mining and cataloged it in the repository, and then it can be modified to be adopted in new software product. Reusable asset with internal components is the software component contains a source code which allows to be modified in order to fit certain software product. The cost of this asset is very high comparing with other assets.

However, the new method of reuse allows developing new reusable assets during software product development which reduce the cost of developing this type of asset. That in this type of development, the cost of develop a reusable asset is equal to the difference between normal development and developing a new asset for reuse.

The external assets (COTS) are used to reduce the time of developing software products, whereas this asset does not allow for modification that caused problems in software development. The problems occurred when the modification is required to fit in the system, instead of modify the asset, the system components are needed to be modified.

Software components operations and transitions are considered in the models. Component transition shows the software components between the products and the repository without any changes in the components. The operation shows the process of modify the software components within software products and the repository. This process allows changing the characteristics of the software components in order to be used in different types of software products.

However, the proposed model divided the process of cataloged and mining software components into two different tasks. That is the new type of reuse requires only one task instead of both. The new method does not require a mining in the software product during development that is the required component is defined early in software development.

The white box reuse consists of two different tasks, retrieve the required components from the repository and adopts the component in the new system. This process has been defined in the previous model as a single task, while in order to differentiate between the process and the assets we divided this process into two different tasks.

The final result of the reuse process is a software component adopted in the new system. The asset it can be adopted and used in the new system can be in three different forms, modified software component, non modified software component, and software component with modified system.

The modified software component is the software asset modified or developed from scratch in order to fit the new software. Only the in house development assets are applicable to be in this state. This asset is restored in the repository as a new software asset or a new version of software asset. The high rate of using this way is by using a normal asset in software reuse. That this type of asset is developed for specific task and condition and it is not developed to be used in other software products.

Non modified software component is a software asset that fit the new requirement without any modification. This type of reuse is compulsory for external assets that it does not allow the modification.

Software component with modified system, in this way the problem of using the external assets is rising. That is the system architecture and components are modified in order to be suitable for the imported asset.

Whereas, different ways of conduct a reuse process. The cost of the reuse process is calculated based on the way of develop, transit, transform, and reuse the software assets. As shows in figure 3, the cost of reuse process is the composition of the cost of different tasks that conducted for one process.

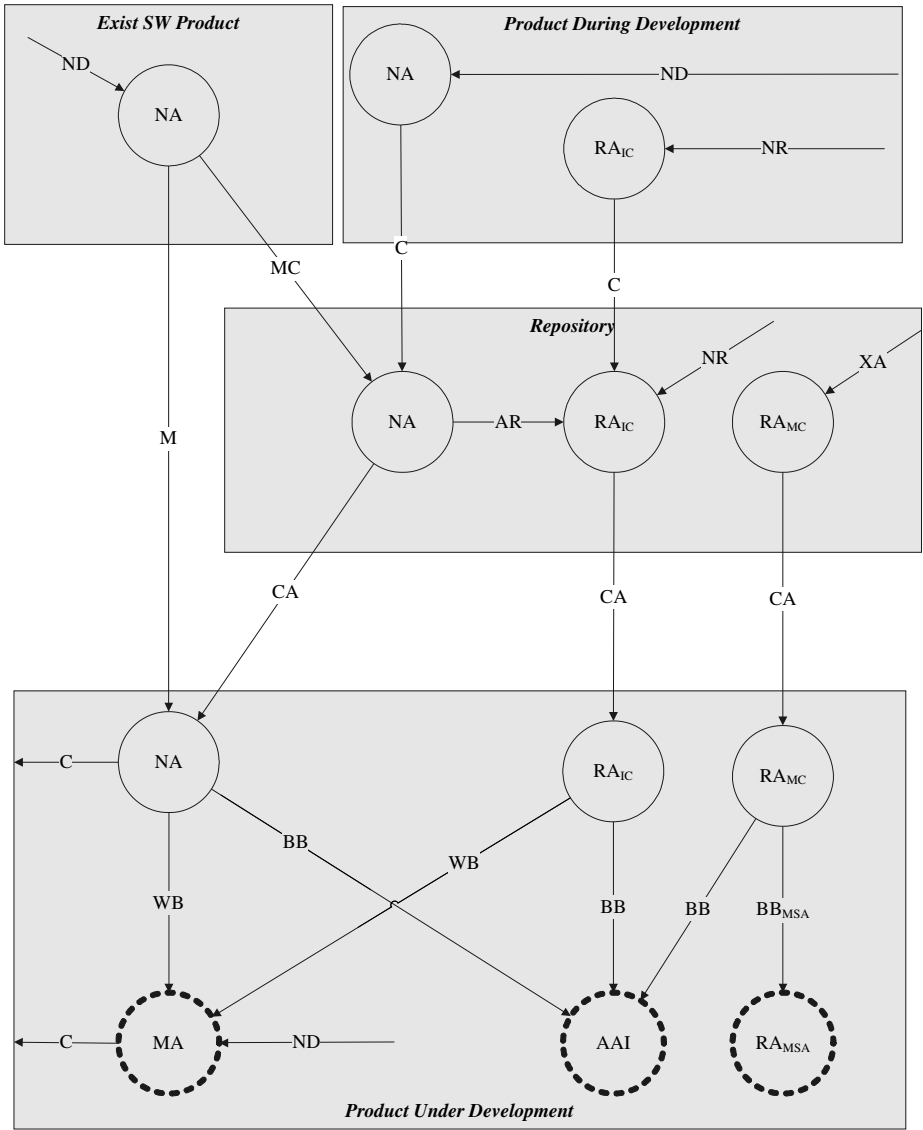


Fig. 3. Cost of Elementary Reuse, Transitions and Transformation

6 Conclusion

The reuse strategy is the most significant strategy used to reduce the cost of software development. That intends to reuse existing software components in order to develop new software products. The problem faced in this strategy is a lack of defined systematic reuse. Different methods of software reuse have been proposed. Whereas the reuse process is completely based on the available software components are able to be used.

Therefore, in this study, we discussed a new method of software reuse that intends to use the process of developing a software product in order to produce reusable components. In this way, two objective are achieved with the lowest cost can be paid. The process of developing reusable software components during software product development required an extra effort in order to consider the reusability characteristics.

A new model of software reuse evaluation is developed in order to calculate the new tasks and to show the different state of the assets in software reuse strategy. In this model, the project manager is able to define the method of software reuse early, instead of follow the available software components when they intend to develop a new software product. Moreover, the developed model presents the applicable ways of software reuse and their cost.

References

1. AL-Badareen, A.B., Selamat, M.H., Jabar, M.A., Din, J., Turaev, S.: Reusable Software Components Framework. In: European Conference of Computer Science (ECCS 2011), pp. 126–130. NAUN, Puerto De La Cruz (2010)
2. AL-Badareen, A.B., Selamat, M.H., Jabar, M.A., Din, J., Turaev, S.: Reusable Software Component Life Cycle. *International Journal of Computers* 5, 191–199 (2011)
3. Nakano, H., Mao, Z., Periyasamy, K., Zhe, W.: An Empirical Study on Software Reuse. In: International Conference on Computer Science and Software Engineering, vol. 6, pp. 509–512 (2008)
4. Zhu, Z.: Study and Application of Patterns in Software Reuse. In: International Conference on Control, Automation and Systems Engineering, pp. 550–553 (2009)
5. Yong, I., Aiguang, Y.: Research and Application of Software-reuse. In: Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, vol. 3, pp. 588–593 (2007)
6. Fadila, A., Mohamed, A.N.: Reusing heterogeneous software process models. In: IEEE Symposium on Computers and Communications, pp. 291–294 (2009)
7. Bellettini, C., Damiani, E., Fugini, M.G.: Software reuse in-the-small: automating group rewarding. *Information and Software Technology* 43, 651–660 (2001)
8. Guo, J.: Software reuse through re-engineering the legacy systems. *Information and Software Technology* 45, 597–609 (2003)
9. Hewett, R.: Learning from software reuse experience. In: International Symposium on Empirical Software Engineering 2005, p. 10 (2005)

10. Boehm, B.: Managing software productivity and reuse. *Computer* 32, 111–113 (1999)
11. Ramachandran, M.: Software reuse guidelines. *SIGSOFT Softw. Eng. Notes* 30, 1–8 (2005)
12. Ravichandran, T., Rothenberger, M.A.: Software reuse strategies and component markets. *Commun. ACM* 46, 109–114 (2003)
13. Tomer, A., Goldin, L., Kuflik, T., Kimchi, E., Schach, S.R.: Evaluating software reuse alternatives: a model and its application to an industrial case study. *IEEE Transactions on Software Engineering* 30, 601–612 (2004)
14. Schach, S., Schach, R., Schach, S., Tomer, A.: *Development/Maintenance/Reuse-Software Evolution in Product Lines*. pp. 437. Springer, Netherlands (2000)

Achieving Effective Communication during Requirements Elicitation - A Conceptual Framework

Fares Anwar, Rozilawati Razali, and Kamsuriah Ahmad

Center Of Software Technology and Management, Faculty Of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor Darul Ehsan
fares1983@hotmail.com, rozila@ftsm.ukm.my, kam@ftsm.ukm.my

Abstract. Requirements elicitation is one of the most important and critical phase in software development. It is the moment in which the users' needs of a software system are captured, understood and validated. This is achieved through two-way communications between users and requirement analysts. The process however is not so straightforward to accomplish. The problem of poor communication among requirement analysts and users exists since both parties are different in many ways besides the environment in which the process happens. They face significant challenges to achieve common understandings and agreements on requirements. This paper discusses the contributing factors that affect communications between both parties during requirements elicitation phase. The factors were identified through reviews of related work. The data were analysed through content analysis. The collated factors form a conceptual framework of effective communication activity for requirements elicitation process. The aim of the framework is to ensure the produced requirements are comprehensible and thus leads to the production of a software system that satisfies its intended users.

Keywords: Requirements elicitation, Communication gaps, Effective communication.

1 Introduction

Requirements elicitation is one of the most important and critical phase in software [1], which have direct influence on software quality and cost [2]. There are two key players during the process, namely users and requirement analysts. The process requires requirement analysts to first knowing what they must create and how to create it. This normally includes determining what functions a system must perform and how well it must accomplish the required functions [3].

Requirements elicitation involves a rich communication activity that entails users to have the ability to interact and communicate their needs to analysts [4],[5]. The analysts on the other hand should have the capacity to mine and grasp the necessary domain knowledge from the users. Communication conflict is inevitable during the process since preference, priorities, backgrounds, objectives and goals vary from one person to another. The users and analysts thus need to have appropriate communication skills to reduce the conflicts [6]. Perhaps only the people who have good communication skills should participate in the process.

There are significant differences in perception between analysts and users. The former prefers to view the system from technological perspective and the later from business perspective. During requirements elicitation, users exchange knowledge about their organisations' business process, activities and work practice that should be considered by analysts when they develop a system. This task is not easy to accomplish as users need to clearly articulate not only explicit but also tacit knowledge to the analysts. In some cases, users are not sure what knowledge that the analysts require. The analysts therefore fail to understand the users' expectations of a system. This hampers the requirements elicitation process to happen effectively, which leads to producing incomplete and inconsistent requirements [7]. Misunderstandings and communication gaps between these two groups can cause negative consequences to software projects as a whole such as cost overruns, extended schedules, fail to satisfy user needs and lower system usability [8],[9],[10],[11],[12].

This paper aims to discuss the contributing factors that affect communication among users and analysts during requirements elicitation process. The factors were identified through literature review, which data were analysed through content analysis. The collated factors form a conceptual framework of effective communication during requirements elicitation process. The paper is organised into four sections. The following section provides the related work on the subject matter that acts as the basis for the proposed framework. Section 3 explains briefly the method used for the data analysis and elaborates the framework. Section 4 provides some ideas on how the framework can be used and applied. Finally, Section 5 concludes the paper with a summary of the main findings and future work.

2 Related Work

Communication is a complex and ongoing process that consists of message exchange between source and receiver through a channel where the source transmits a message to the receiver and the receiver in return provides the necessary feedback [13],[14]. The process is constrained by certain aspects. Figure 1 and Table 1 below depict the communication process and components respectively. If any of these components does not exist as the way it should be, a communication gap is believed to happen. A communication gap is defined as an interaction subject to distortion during information transfer due to physical distance or variation in skills, experiences and backgrounds of communicating people [11].

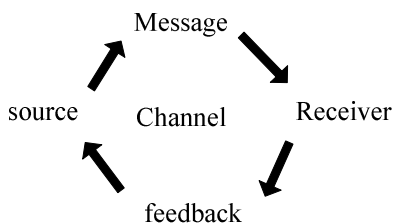


Fig. 1. Communication process between two parties

Table 1. The components of the communication process during requirements elicitation phase

Components	Variables description
Message	Input (user requirements that needed by system analysts to develop system.)
Source and Receiver	Personalities(characteristics that IS professionals and users brought with them during interaction process)
Channel	Medium (the menu that will be used to transfer requirements between the two groups)
Constraints	Communication skills (the capability and skills that IS professionals need to have to improve the communication with other parties).

A number of studies have been conducted to understand and address the communication gaps during requirements elicitation process. For example, several studies have found that individual issues such as educational background and experience that analysts and users have brought during requirements elicitation have direct effects on the interaction [15],[16]. Education in particular affects a teams' ability to explore new options for investigating and gathering requirements [17], as the members may not have a common understanding of the terms used [15].

An Influence on Consensus (IC) Model was proposed by Price and Cybulski, which focuses on the impact of communication factors on stakeholder's consensus during requirements negotiation [5]. The purpose of IC model is to solve conflicts to reaching agreement and consensus between stakeholders during requirements elicitation from communication perspective. The study showed that factors such as trust, knowledge, power, culture and collaboration contribute some levels to communication process. Trust in particular affects the consensus between analysts and users and has a direct relation with knowledge and could not occur without it. Trust plays an essential role in knowledge transfer. To understand domain problems and elicit requirements effectively, the level of trust between analysts and users needs to be high [18]. In other words, mutual trust among analysts and users need to occur [19].

Some studies explored communication problems during software development by examining the gaps between expectation and performance perception on communication skills. The failure to meet expectations of user's perception on skill performance has a significant impact on user's satisfaction and job performance [20]. The perception between analysts and users on the importance and performance of communication skills include both written and oral skills during interaction. Studies highlighted that there are differences in perceptions on the importance and proficiency levels for written and oral communication skills [21],[10]. One problem that becomes apparent during requirements elicitation that leads to poor communication is when analysts start using technical jargons. This may force users who have less technical background to hide some of the requirements and cause misinterpretations [22]. The analysts thus need appropriate oral and writing skills to avoid poor communication. This may imply that project managers need to select a qualified development team to capture and convey ideas and notions clearly and unambiguously to the stakeholders [23].

In addition, the right selection of people and elicitation methodology is also critical. The way how a communication is being handled influences the adoption and selection of elicitation methodology [24]. On the other hand, the success degree of using the methodology depends on the nature and limitation of the organisation that adopts it. An appropriate selection of stakeholders is important in order to understand an organisation's environment. With a clear understanding of organisation, software will be developed based on high quality requirements specifications.[25]. Moreover, stakeholder's identification during requirements elicitation process is crucial as the relationship and communication behavior between diverse groups starts at this phase. Selecting inappropriate people will limit the contribution in communication and consequently, the quality of the system to be built may be jeopardised. Appropriate candidates who are expected to participate in an active and direct manner should be chosen to establish workable communication links. The lack of clear criteria in identifying and selecting ideal candidates to share and gather facts is one of the factors that obstruct effective communication during elicitation [26]. To ensure appropriate stakeholders are selected, criteria such as role establishment, skills analysis and allocation of requirements priorities may be used [25].

The organisational culture can also be a reason why user requirements are not captured into appropriate detailed level. For example, some requirements elicitation techniques such as Joint-Application-Development (JAD) are not appropriate to gather requirements in some situations because of the difference in culture between analysts and users [10]. In fact, culture values in terms of resistance to adapt a new technology may also affect the implementation of the technology within the organisation [2]. Users resistance will lead to unwillingness to involve in requirements elicitation process. This will force analysts to choose other users who may not have appropriate knowledge and eventually affect the quality of the system at the end [27]. Besides technological aspect, there are other types of resistance such as time resistance, overload resistance and silence resistance, which must be addressed adequately [28].

It is well known that documentation can be a medium of communication between analysts and users. However, a study has found that document is a very ineffective option and poor replacement for interpersonal communication skills [4]. Interpersonal skills are necessary to interact with potential stakeholders to produce a clear requirement document and to express ideas effectively [20]. The skills are crucial for completing software projects [1]. Therefore, there is a need to focus on improving interpersonal skills such as collaboration and negotiation besides technical skills in order to make effective communication [29].

Negotiation is one of the important interpersonal skills in requirements elicitation, which deals with conflicts among stakeholders. Since analysts and users have different concerns, priorities and responsibilities, negotiation is necessary in handling conflicts in order to gain better requirements [5],[30]. Because of that, the adoption of negotiation is relatively high (89.58%) in most organisations where it can be used to define system boundaries and priorities [31]. For an effective communication to happen during requirements elicitation, common understanding among different groups is needed, which can only be established through negotiation [32].

Collaboration plays an important role during elicitation process. An effective collaboration will avoid traps where the projects may encounter at the later phase

[33]. Requirements elicitation by nature is a large collaborative task that depends on effective interaction and communication between participants [30]. When multiple stakeholders are participated in the process, requirements may conflict because of different backgrounds among participants who use different terms that makes the collaboration within teams difficult [34]. Collaboration also influences analysts' and user's conscience [5].

Developing a system from incomplete and unclear requirements leads to not meeting user needs and affecting the usability of the system. In order to avoid such a problem, analysts and users need to share the same knowledge during elicitation process. Knowledge is divided into tacit knowledge and explicit knowledge [35]. An effort has been given to provide a medium for communication to increase shared understanding and reduce tacit requirements between analysts and users through a technique called "regrid" [1]. However, this technique contains some drawbacks such as slow, complex, unfamiliar structure and needs more cognitive effort between analysts and users. To overcome the limitations, one suggested increasing common knowledge between analysts and users through putting system developers into user's environment domain [9]. Knowledge integration of both analysts and users have direct impacts on performance and success of the final project [36]. An effective communication starts when common knowledge happened among them.

Based on the reviews above, it can be seen that various factors that affect effective communication during requirements elicitation process have been identified by previous studies. However, most of the factors were identified individually and thus they are narrowed and isolated. It is unclear how these factors are interrelated with each other. This paper is intended to integrate these factors conceptually as a framework so that their effects on the matter can be clearly seen. This later can direct more fruitful research work.

3 The Framework

This paper aims to answer the following questions:

What are the contributing factors that affect communication among users and analysts during requirements elicitation process? How are those factors interrelated?

In order to answer the questions, this study employed content analysis. Content analysis is a summarising and analysis of messages that relies on scientific method. It is a systematic, replicable technique for compressing many words of text into fewer content categories based on explicit rules of coding [37]. It can be used to generate hypotheses of a phenomenon. This is achieved through identification of categories, which in fact, is the important step in content analysis. The hypotheses comprise a set of abstract categories that are systematically connected through certain statements of relationship. Based on the content analysis made on previous studies, several categories of factors that affect communication among analysts and users during requirements elicitation process have been identified. The categories are integrated in a framework that relates those abstract concepts together. Fig. 2 illustrates the conceptual framework.

The framework consists of three essential categories; *people, process and product*. Each category comprises a set of factors or elements. The first category covers people aspect that includes *human factors*. The second category is *process* that consists of *before* and *during* activities concerning requirements elicitation. The *product* is the third category, which portrays the output of the *process*. The aim of the *process* is to produce comprehensible requirements, which later could satisfy users through the accomplishment of acceptable software. This framework shows how these categories can contribute to effective communication during requirements elicitation process. The details of the factors are described in the following paragraphs.

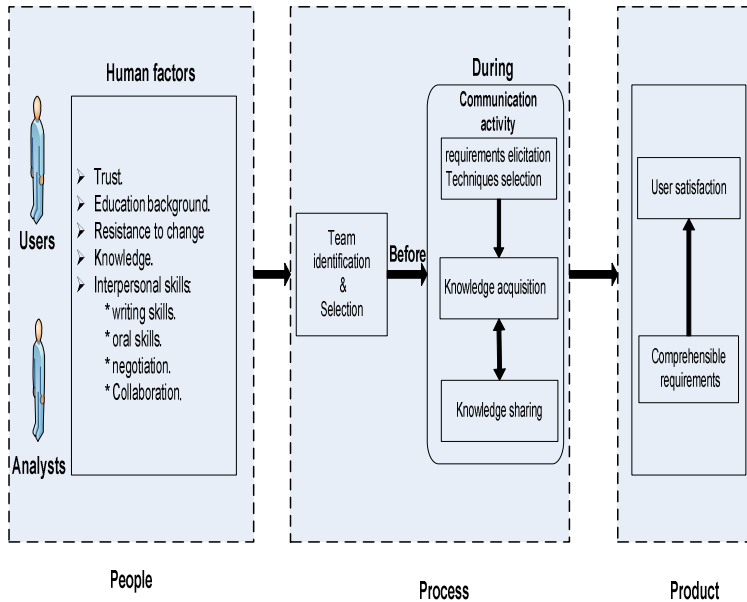


Fig. 2. The contributing factors and elements of effective communication during requirements elicitation process

3.1 People

Requirements elicitation is known as the phase that needs intensive communications among users and analysts to gather the right requirements. It is imperative to focus on *human factors*, which are brought by users and analysts with them during interactions. The factors can be classified into five aspects: trust, education, resistance to change, knowledge and interpersonal skills. The identification and consideration of these human aspects are essential for selecting the appropriate participants to be involved in requirements elicitation process. It is believed that by selecting participants with appropriate intellectual properties (interpersonal skills, education background and knowledge) together with right attitudes (trust and willing to change) may facilitate effective communication. This later leads to better requirements capturing.

3.2 Process

This category consists of two parts: *before* and *during*. The *before* activity is mainly the identification and selection of participants whereas the *during* activities comprise elicitation technique selection, knowledge acquisition and sharing [24],[26]. The former is accomplished by considering the characteristics of the people involved, as mentioned above. The latter is the significant part because it concerns the *communication activity* by which common understanding should be achieved. The activity requires user participation. It is an exchange of formal and informal information between users and analysts. Knowledge is difficult to acquire and share during this phase because the people involved come from distinct backgrounds with different way to store, percept and express their knowledge about the requirements. Suitable techniques have to be identified in order to ensure the activity can be accomplished successfully. The techniques will act as the catalyst that encourages users to transfer and share their business knowledge to analysts. The analysts on the other hand will share their technical knowledge to users so that comprehensible requirements can be produced. The diversity of stakeholders with different personalities and ways in expressing knowledge normally imposes the use of more than one elicitation method. Some examples of techniques include synthetic (JAD, RAD, Prototype) that allows analysts and users communicate collaboratively, analytic (Repertory grid, documentation studies, requirements reuse) that captures requirements in indirect ways, observational (ethnography study, protocol analysis) that acts as a means to understand application domain and tacit requirements, and conversational (interviews, focus groups, brain storming) that works best at capturing non-tacit or explicit requirements [38].

Knowledge acquisition can be defined as the need to establish connections between analysts and users in terms of experience, business and technical knowledge [26]. This connection needs to be introduced in order to develop a shared or mutual understanding and common views of desired future software [24]. The main barrier that hampers knowledge acquisition is inadequate understanding between users and analysts. Requirements elicitation in essence is the process of knowledge sharing that happens in two flows. The first knowledge flow is from users to analysts, which is knowledge in business domain and users' tacit knowledge. The second knowledge flow is from analysts to users, which is the knowledge in software domain and system performance requirements [35]. The main objective for this stage is to define requirements through thorough understanding between users and analysts perspective. A shared perspective needs to take place between both parties.

3.3 Product

Reaching common understanding among users and analysts is essential to get comprehensible requirements. Comprehensible means that all readers of requirements are able to understand what each requirement is saying and reach single consistent interpretation of it [39]. Comprehensible requirements are necessary to ensure that the system models produced are easy to validate by users. Validated models could confirm that the software to be built would satisfy users as it meets what they want. It is not doubt that there are many representations that can be used in requirements

specifications. But, not all of them can be easily be understood by users and analysts. Some representations are more comprehensible than others [40], while some of them can facilitate understanding better if they are combined [41]. Therefore, a representation that is user-friendly should be identified and suggested.

4 Applying the Framework

The framework may not be conclusive as it needs to be confirmed and refined through empirical settings. However, it provides a conceptual overview on how those factors are interrelated with each other in the effort to achieve effective communication during requirements elicitation process. In particular, the framework can be used by practitioners as a guide to perform the elicitation process. For example, they are now aware of the importance of considering human factors in selecting the right participants and elicitation techniques. Attention therefore must be given not only during but also before the requirements elicitation process. From the framework, practitioners also understand that requirements elicitation is indeed knowledge acquisition process. Therefore, organisations must encourage and support knowledge sharing culture.

To researchers, the framework triggers the needs to identify the suitable mechanism that can help practitioners in executing the above process. Methods, techniques or models have to be developed to support the selection and knowledge acquisition processes. In short, below are some research questions that future research may focus on:

- *How the team members (participants) for requirements elicitation process can be selected appropriately?*
- *How to determine which requirements elicitation techniques are suitable for which environment and should be used by whom (characteristics of participants)? What would be the selection criteria?*
- *How can the concepts of knowledge management be adopted in requirements elicitation process?*

5 Conclusions and Future Work

This paper discussed the importance of having an effective communication during requirements elicitation, as any mistake made during this phase may increase cost and lead to system and project failures. Several contributing factors and elements for an effective communication have been identified from the literature. The factors were identified through content analysis which later were grouped into three categories; people, process and product. The factors together with the categories form a conceptual framework of effective communication during requirements elicitation process. The framework can guide future research by highlighting the important elements that need further investigation. For instance, future studies may explore the methods that can assist in selecting team members and stakeholders as well as

elicitation techniques used during the process. A mechanism that encourages knowledge sharing and reduces the gaps that hamper knowledge acquisition may also be introduced. In addition, representations that can promote common understanding could also be proposed.

References

1. Davis, C., Fuller, R., Tremblay, M., Berndt, D.: Communication challenges in requirements elicitation and the use of the repertory grid technique. *Journal of Computer Information Systems* 46, 78–86 (2006)
2. Cybulski, J., Sarkar, P.: Requirements engineering for web-based information systems. *Engineering and managing software requirements*, 327–349 (2005)
3. McAllister, C.A.: Requirements determination of information systems: User and developer perceptions of factors contributing to misunderstandings. Unpublished doctoral dissertation, Capella University (2006)
4. Al-Rawas, A., Easterbrook, S., Aeronautics, U.S.N., Administration, S.: Communication problems in requirements engineering: a field study, Citeseer (1996)
5. Price, J., Cybulski, J.: Consensus Making in Requirements Negotiation: the communication perspective. *AJIS* 13 (2005)
6. Saedian, H., Dale, R.: Requirements engineering: making the connection between the software developer and customer. *Information and Software Technology* 42, 419–428 (2000)
7. Laporti, V., Borges, M., Braganholo, V.: Athena - A collaborative approach to requirements elicitation. *Journal of Computers in Industry*, 367–380 (2009)
8. Mann, J.: IT education's failure to deliver successful information systems: Now is the time to address the IT-user gap. *Journal of Information Technology Education* 1, 253–267 (2002)
9. Friedrich, W., Van Der Poll, J.: Towards a methodology to elicit tacit domain knowledge from users. *Interdisciplinary Journal of Information, Knowledge, and Management* 2, 179–193 (2007)
10. Tuffley, D.: Exploring the IT-User Gap: towards developing communication strategies. *QualIt: challenges for qualitative research* (2005)
11. Kumlander, D.: Communication gaps and requirements uncertainties in the information systems design. *World Scientific and Engineering Academy and Society (WSEAS)*, 400–405 (2006)
12. Gissel, R.L.: Information system requirements determination: Factors impeding stakeholders from reaching common understandings and agreements on requirements (2010)
13. Galvin, K., Wilkinson, C.: The communication process: Impersonal and interpersonal. *Making connections: Readings in relational communication* 3, 4–10 (2003)
14. Zin, A., Pa, N.: Measuring communication gap in software requirements elicitation process. In: *World Scientific and Engineering Academy and Society (WSEAS)*, pp. 66–71 (2009)
15. Kotonya, G., Sommerville, I.: Requirements engineering with viewpoints. *Software Engineering Journal* 11, 5–18 (1996)
16. Urquhart, C.: Analysts and clients in organisational contexts: a conversational perspective. *The Journal of Strategic Information Systems* 10, 243–262 (2001)

17. Cybulski, J., Systems, D.U.S.: Understanding problem solving in requirements engineering: debating creativity with IS practitioners. Deakin University School of Information Systems (2003)
18. Chakraborty, S., Sarker, S.: An Exploration into the Process of Requirements Elicitation: A Grounded Approach. *Journal of the Association for Information Systems* 11, 1 (2010)
19. Hickey, A., Davis, A.: Elicitation technique selection: how do experts do it? In: *International Conference on Requirements Engineering*, pp. 169–178. IEEE, Los Alamitos (2003)
20. Hornik, S., Chen, H., Klein, G., Jiang, J.: Communication skills of IS providers: An expectation gap analysis from three stakeholder perspectives. *IEEE Transactions on Professional Communication* 46, 17–34 (2003)
21. Chen, H., Miller, R., Jiang, J., Klein, G.: Communication skills importance and proficiency: perception differences between IS staff and IS users. *International Journal of Information Management* 25, 215–227 (2005)
22. Alkadi, G., Beaubouef, T., Schroeder, R.: The sometimes harsh reality of real world computer science projects. *ACM Inroads* 1, 59–62 (2010)
23. Vale, L., Albuquerque, A., Beserra, P.: Relevant Skills to Requirement Analysts According to the Literature and the Project Managers Perspective. In: *International Conference on the Quality of Information and Communications Technology*, pp. 228–232. IEEE, Los Alamitos (2010)
24. Coughlan, J., Macredie, R.: Effective communication in requirements elicitation: A comparison of methodologies. *Requirements Engineering* 7, 47–60 (2002)
25. Pacheco, C., Garcia, I.: Effectiveness of Stakeholder Identification Methods in Requirements Elicitation: Experimental Results Derived from a Methodical Review. In: *International Conference on Computer and Information Science*, pp. 939–942. IEEE, Los Alamitos (2009)
26. Coughlan, J., Lycett, M., Macredie, R.: Communication issues in requirements elicitation: a content analysis of stakeholder experiences. *Information and Software Technology* 45, 525–537 (2003)
27. Thanasankit, T., Corbitt, B.: Cultural context and its impact on requirements elicitation in Thailand. *The Electronic Journal of Information Systems in Developing Countries* 1 (2000)
28. Hayat, F., Ali, S., Ehsan, N., Akhtar, A., Bashir, M., Mirza, E.: Requirement elicitation barriers to software industry of Pakistan (impact of cultural and soft issues). In: *International Conference on Management of Innovation and Technology (ICMIT)*, pp. 1275–1278. IEEE, Los Alamitos (2010)
29. Hall, T., Wilson, D., Rainer, A., Jagielska, D.: Communication: the neglected technical skill? In: *ACM SIGMIS CPR conference on Computer personnel research*, pp. 196–202. ACM, New York (2007)
30. Ahmad, S.: Negotiation in the requirements elicitation and analysis process. In: *19th Australian Conference on Software Engineering*, pp. 683–689. IEEE, Los Alamitos (2008)
31. Solemon, B., Sahibuddin, S., Ghani, A.: Adoption of Requirements Engineering Practices in Malaysian Software Development Companies. *Advances in Software Engineering*, 141–150 (2010)
32. Erra, U., Scanniello, G.: Assessing communication media richness in requirements negotiation. *Software, IET* 4, 134–148 (2010)
33. Wieggers, K.: Karl Wieggers describes 10 requirements traps to avoid. *Software Testing & Quality Engineering* 2 (2000)

34. Fuentes-Fernández, R., Gómez-Sanz, J., Pavón, J.: Understanding the human context in requirements elicitation. *Requirements Engineering*, 1–17 (2010)
35. Wan, J., Zhang, H., Wan, D., Huang, D.: Research on Knowledge Creation in Software Requirement Development. *Journal of Software Engineering and Applications* 3, 487–494 (2010)
36. Tesch, D., Sobol, M., Klein, G., Jiang, J.: User and developer common knowledge: Effect on the success of information system development projects. *International Journal of Project Management* 27, 657–664 (2009)
37. Weber, R.P.: *Basic Content Analysis*, Newbury Park, CA (1990)
38. Zhang, Z.: Effective Requirements Development—A Comparison of Requirements Elicitation Techniques. In: Berki, E., Nummenmaa, J., Sunley, I., Ross, M., Staples, G. (eds.) *British Computer Society*, pp. 225–240 (2007)
39. Wiegers, K.E.: *Software requirements*. Microsoft Press Redmond, WA (2003)
40. Razali, R., Snook, C.F., Poppleton, M.R., Garratt, P.W., Walters, R.J.: Experimental Comparison of the Comprehensibility of a UML-based Formal Specification versus a Textual One. In: *International Conference on Evaluation and Assessment in Software Engineering (EASE)*, British Computer Society, pp. 1–11 (2007)
41. Razali, R., Najafi, P., Mirisae, S.H.: Combining Use Case Diagram and Integrated Definition's IDEFO—A Preliminary Study. In: *International Conference on Software Engineering and Data Mining (SEDM)*, pp. 231–236. IEEE, Los Alamitos (2010)

Investigating the Effect of Aspect-Oriented Refactoring on Software Maintainability

Hamdi A. Al-Jamimi, Mohammad Alshayeb, and Mahmoud O. Elish

Information and Computer Science
King Fahd University of Petroleum and Minerals
Dhahran 31261, Saudi Arabia
{aljamimi, alshayeb, elish}@kfupm.edu.sa

Abstract. Refactoring improves software quality by improving the design of existing code through changing its internal structure while preserving its behavior. A number of refactorings were proposed specifically for Aspect-Oriented (AO) systems. AO techniques are emerging to cope with the challenges of current software development and to address shortcomings of existing paradigms. Each of the proposed aspect-oriented refactorings (AOR) has a particular purpose and effect, thus their effect on software quality attribute may vary. The software maintenance activities cost lots of time and effort to be performed. In this paper, we propose a classification of AOR based on their measurable effect on software maintainability. The aim of the classification is to help the software designer and developer decide which AOR can be applied to optimize AO system with respect to maintainability.

Keywords: aspect-oriented, refactoring, aspect-oriented refactoring, software metrics, maintainability.

1 Introduction

The software system's purpose is to provide some functionality to its users. However, the software system can not merely be concerned with its main purpose. In many cases, the system must also provide additional concerns, which are outside the main domain, to complement its main function, as well as to produce a successful, robust and complete system [1]. Adding like these secondary concerns and changing in requirements make the code a subject for modifications and improvements.

Hence, instead of cluttering up the structure the changes can be accomplished by applying another kind of changes, called refactorings, improving only the internal structure of the system without altering its behavior [2, 3]. Refactorings generally represent transformations that may be applied to code given with considering some restrictions to guarantee that the refactoring maintains program consistency and preserves its behavior.

Aspect-Oriented Software Development (AOSD) [4] is an emerging paradigm that comes to complement object-orientation. AOSD helps developers to solve problems related to modularity that could not be properly addressed by OO software.

Aspect-orientation focuses to enable the clean modularization of crosscutting concerns such as security, logging, exception handling and others in a more effective manner [5, 6].

AOR differs from traditional refactorings in that they involve aspect-oriented programming (AOP) constructs (such as advice, pointcut, introduction..) either as the targeted elements or in the resulting code. In general, most refactoring methods increase the modularity of code and eliminate redundancies. The same improvements are gained by AOSD, so it is normal to apply refactoring and AOP within the same development process.

The maintenance activities of the AO software system are critical issue since they cost a lot of time and efforts to be accomplished [7]. Software maintainability as defined by ISO 9126, *is as the ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment* [8]. So high maintainable software demands less rework and thus is overall less costly. Software maintenance is one of the external quality attributes that can not be assessed directly, so software metrics are the means to assess it [9, 10].

The objective of this paper is to propose a classification of AOR based on their measurable effect on software maintainability to help the software developers decide which AOR to apply in order to enhance the maintainability of the AO system.

The rest of the paper is organized as follows; Section 2 summarizes related work, while Section 3 shows the software maintainability assessment. The investigated AOR are introduced in Section 4. In Section 5 we detail the research method. Then, the effects of AOR on the software quality attributes are discussed in Section 6, where a discussion of the results is detailed in Section 7. Finally, the conclusions and future work are stated in Section 8.

2 Related Work

There are many studies that investigated the effects of AOP on the software quality attributes. Zakaria et al. [11] discussed the effects of aspects on C&K metrics suite. They found that all the metrics in the suite might be affected in some way. Kvale et al. [12] studied how AOP ease the adding and replacement of components in COTS-based development. They re-engineered an existing OOP application using AOP and compared the LOC and number of classes needed to be changed in order to add and replace COTS components. Results from their study showed that proper use of AOP in COTS component integration can help to increase the changeability of the system. Madeyski et al. [13] carried out an empirical study of a web-based system to examine AO vs. OO approach with regard to software development efficiency and design quality. The study revealed that the AOP approach appears to be a full-fledged alternative to the pure OO approach.

Moreover, Tonella and Ceccato [14] conducted a study that focused on a specific kind of crosscutting concerns, called aspectizable interfaces. All the aspectizable interfaces identified within a large number of classes from the Java Standard Library and from three Java applications have been automatically migrated to aspects. They focused on two internal attributes: size and modularity. They performed some maintenance tasks on the two alternative versions (with and without aspects) of the

same system. The results indicated that among the internal attributes, only those referred to the modularity had a significant change where the size was not significantly affected. For the external attributes, they concluded that the AO code became easier to understand. To some extent, it also became easier to maintain, although the small size of the affected code portion might have reduced the overall benefits on maintainability.

Kulesza et al. [15] presented a quantitative study that assesses the positive and negative effects of AO programming paradigm on typical maintenance activities of a web information system. The study consisted of a systematic comparison, at the system level, between the OO and AO versions of the application. A suite of metrics for separation of concerns (SoC), coupling, cohesion and size were used. The AO implementation exhibits better results for SoC measures and many other metrics, such as LOC, Number of Attributes (NOA) and the coupling metrics (CBC and DIT). On the other hand, the OO implementation brings better results for the vocabulary size and cohesion metrics. Both implementations present similar results for the Weighted Operations per Component (WOC) metric.

Sant'Anna et al. [16] presented a quantitative comparison between AO and conventional Multi-agent Systems (MAS) architectures. They quantitatively evaluated the degree to which aspect-oriented MAS architectures scale up to promote improved modularity when compared to MAS architectures based on conventional patterns, such as mediator-based and publisher-subscriber styles. The assessment is mainly concerned with the degree with which the modularity supports the adaptability and variability of MAS features.

We can observe the limitations of the existing research studies reviewed above. Only one study [14] investigated the effects of applying kind of AOR on the internal and external quality attributes. Furthermore, none of the existing research studies has classified AOR based on their effects on the internal and external software quality attributes. Thus, in this study we focus on classifying the AOR based on their effects on the software maintainability.

3 Software Maintainability Assessment

The internal software quality attributes can be used to assess the external ones. Software metrics have been used as indicators for external software quality attributes either in OO or AO systems [17-19]. For instance, Sant'Anna et al. [18] defined new metrics suite for AO systems, which are for measuring cohesion, coupling and size related attributes. This metrics suite further has been used to evaluate quality characteristics, which are not directly measurable and can be used to assess reusability and maintainability of AO software. They refined classical Chidamber & Kemerer (C&K) metrics [20, 21], namely: Depth of Inheritance (DIT), Coupling Between Components (CBC), Lack of Cohesion in Operations (LCOO) and Weighted Operations per Component (WOC). Additionally, two other metrics have been used to measure the module size such as Number of Attributes (NOA) and Lines of code (LOC). Their work was evaluated in the context of two different empirical studies with different characteristics such as diverse domains, varying control levels and different degree of complexity. The metrics DIT, CBC, WOC, NOA and LOC were found to be inversely proportional to the software maintainability.

3.1 Aspect-Oriented Metrics

A suite of AO metrics have been used in order to assess the maintainability of the AO software system. More specifically, we consider the extended C&K metrics that were adapted to be applicable to the AO software [18, 22]. The C&K metrics suite provides the most comprehensive and best validated set of measures. Thus, in addition to their widely uses to assess the quality attributes in OO systems, they are used in several studies focusing in the AO systems quality [18, 22-24]. In this study, we used most of these metrics such as DIT, CBC, LCOO and WOC. Additional metrics were used to measure the aspect size such as NOA and LOC. We chose these metrics because they have been used by previous empirical study to assess software maintainability and they were found to be good indicators for maintainability [18]. The AO metrics we investigated are the following:

Table 1. The used AO metrics

Metric	Description
DIT	The length of the longest path from a given aspect to the aspect hierarchy root.
CBC	The number of components or interfaces declaring methods or fields that are possibly called or accessed by a given aspect.
LCOO	It measures the lack of cohesion of an aspect in terms of the amount of operations pairs that don't assess the same instance variable.
WOC	It counts number of operations (advices and methods) in a given aspect.
NOA	It counts the internal vocabulary and the number of attributes of each aspect. The inherited attributes are not included in the count
LOC	It defines as the total source lines of code in a module excluding all blank and comment lines

4 Selected AO Refactoring

Monteiro et al. [25, 26] introduced a collection of refactoring for the AspectJ programming language. Each one includes the motivation of why the refactoring should be performed and step-by-step description of how to carry out the refactoring. In addition, many other AOR have been proposed and used [27-29]. Among the proposed AOR we selected the refactorings that are specific to the aspect code, i.e. do not affect the classes' code. The AOR were selected to cover the different AO constructs such as advice, intertype-declaration, etc. In our study, we investigated the effects of specific refactorings for AO constructs and we divided them into two categories. The first group includes five different AOR which can be applied within aspects. The second group comprises also five AOR to be applied between different aspects.

4.1 AOR within Aspects

- **Change Advice Kind from Around:** is used to separate an advice into multiple advices or into a single less general advice [28].

- **Extract Method from Advice:** is applied if there is a code fragment in an advice body that can be grouped together. Its idea is to move the code fragment into its own method and choose a name for the method to explain its purpose [28].
- **Merge Advice Bodies:** is used when two before or after advice bodies, with the same parameters, are woven to the same joinpoint. As a result, the two advices can be combined together by moving one's body block into the other's body block [27].
- **Generalize before or after Advice to around Advice:** is applied when two different types of advice (before and after) use the same pointcut that means the two advices will be executed at the same joinpoint. Accordingly, the two advices can be generalized together to form one around advice[27].
- **Change Advice kind from Before to After:** is used to change advice kind from before to after or vice versa. For instance, sometimes it is more natural to execute the advice after a particular joinpoint instead of before [28].

4.2 AOR between Aspects

- **Pull Up Advice:** When all sub-aspects use the same advice acting on a pointcut declared in the super-aspect, we can move the advice to the super-aspect [26].
- **Push Down Inter-type Declaration:** is used when an inter-type declaration would be best placed in a sub-aspect rather than in the super-aspect [26].
- **Eliminating Borrowed Pointcut:** is applied when a pointcut is referred by advices of the aspects that are not sub-aspects [29].
- **Eliminating Duplicated Pointcut:** When pointcuts collect the same set of joinpoints in base code, we can apply this refactoring to eliminate the duplications [29].
- **Move Static Introduction:** is applied to move a static inter-type member introduction to a different aspect [27].

5 Research Method

In this section the research method to investigate the effect of selected AOR on software maintainability is presented. In this work, we instigated the effect of AOR on AO system at the aspect level. The motivation behind it is that a single aspect can contribute to the implementation of a number of methods, modules, or objects, increasing or reducing maintainability of the code.

5.1 Data Collection

This section describes the methodology followed to collect data when applying the AOR. The used methodology comprises six steps explained in the following [30]:

1. Collect the internal quality metrics for the target aspect before applying the refactoring. The AO metrics were partially collected by using AJATO tool [31].
2. Perform the AO refactoring.
3. Collect the internal quality metrics after applying the AO refactoring.

4. Report the changes in the internal quality metrics for each aspect in the system.
5. Map the changes in the internal quality metrics to the external quality attribute.
6. Repeat the previous steps for all the investigated AOR.

5.2 Software Systems Background

We have used six open-source AO systems. These software systems have been used for different purposes related to AOP. In addition, they have been chosen based the availability of the systems that have aspects implemented by using AspectJ. The open source projects are: Prevayler [12], AJHotDraw [32], AspectTetris [33], Telecom [34], AJEFW [35], and SapceWar [34].

Summary of the main characteristics of the used software projects in this validation is presented in Table 2.

Table 2. Characteristics of studied software projects

Project	Language	# Classes	# Aspects	Description
Prevayler	Java & AspectJ	90	55	Main memory database system
AJHotDraw	Java & AspectJ	290	31	An open-source drawing application
AspectTetris	Java & AspectJ	7	8	A game developed at Blekinge Institute of Technology
AJEFW	Java & AspectJ	18	4	A program for the treatment of different kinds of exceptions present in application
Telecom	Java & AspectJ	10	3	A simple communication program
SapceWar	Java & AspectJ	15	4	A simple game

5.3 Example

In this section we introduce an example of how to investigate the effect of AOR on the software maintainability using the "*Move Static Introduction*" refactoring [30].

This refactoring technique is used to move a static inter-type member introduction to a different aspect, we can inline (localize) it from the original aspect into the target one. In Figure 1 and 2 the idea of this AO-refactoring technique is explained by diagrams which might be more understandable than a long source code.

However, *aspectB* implements a concern that is relevant to *ClassD* and it introduces the needed attributes to the target class. Thus, the declared *introD2* can be moved from *aspectA* to *aspectB* since it is relevant to the concern of *aspectB*.

Before the "*Move Static Introduction*" is applied, the internal quality metrics for the above example (i.e. for the original aspect *aspectA*) are collected. They are collected for the same aspect again after "*Move Static Introduction*" is applied. This enables us to observe the changes in the internal quality metrics caused by applying this AO-refactoring method. As a result, the coupling, cohesion and the size of *aspectA* are decreased. It does not have an effect on the rest of the investigated AO metrics.

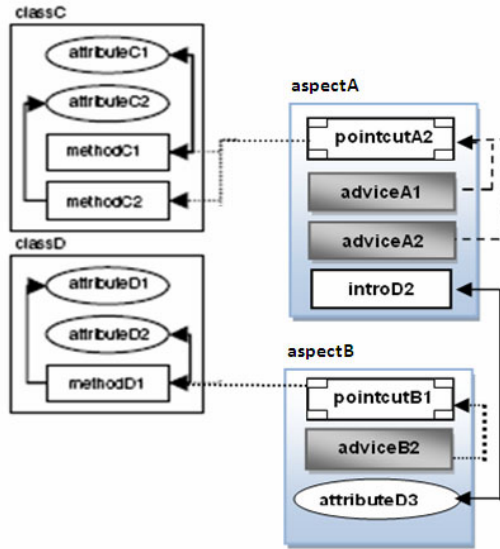


Fig. 1 Before applying “Move Static Introduction”

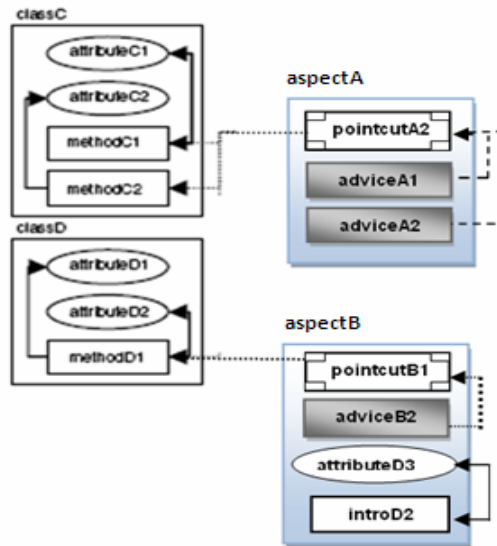


Fig. 2. After applying “Move Static Introduction”

To demonstrate the effect of “*Move Static Introduction*” on the software maintainability, we map the changes in the internal quality metrics to the maintainability attribute based on [18]. For example, to see whether “*Move Static Introduction*” improves the software maintainability or not, we can observe from the previous example that “*Move Static Introduction*” decreases the metrics value of

CBC, LCOO, WOC, and LOC. The changes in these metrics values are mapped to the maintainability attribute based on [18] that shows a negative correlation between the maintainability and CBC, LCOO, WOC, and LOC metrics. Therefore, "*Move Static Introduction*" increases (improves) the software maintainability as the coupling, cohesion and size of the aspect metrics decrease.

6 Affects of AOR on the Software Quality Attributes

This section demonstrates the changes occurred in the internal and external quality attributes when applying AOR. In the first subsection, we explain the changes occurred on AO metrics when applying each AOR. Then, the subsection 6.2 demonstrates the effects of AOR on software maintainability by mapping the changes in the internal quality attributes to maintainability.

6.1 AOR Effects on AO Metrics

The changes in the studied internal quality metrics, caused by applying the investigated AOR, are shown in Table 3. The occurred changes are concluded from the empirical results from all studied software projects. In the following we analyze the effect of each of the AOR on the studied internal quality metrics. That is, to give details why the internal quality metric increases or decreases as a result of applying a particular AO-refactoring.

- **Change Advice Kind from Around:** does not affect coupling and depth of inheritance metrics since it neither inherits an aspect nor use additional methods or joinpoints of other aspects or classes. However, it increases the values of LCOO, WOC and LOC metrics since it separates a single advice into two advices and each of the new advices needs its own signature. In summary, *Change Advice Kind from Around* increases the aspects size in terms of WOC and LOC and makes the aspect less cohesive as it assigns more responsibilities to it.
- **Extract Method from Advice:** does not affect DIT and CBC metrics since it neither inherits an aspect nor use additional methods or joinpoints. However, it increases the values of LCOO; WOC and LOC metrics since it introduce a new method. In summary, *Extract Method from Advice* increases the aspect size in terms of WOC and LOC and makes the aspect less cohesive as it assigns more responsibilities to it.
- **Merge Advice Bodies:** does not affect coupling and depth of inheritance metrics since it neither inherits an aspect nor use additional methods or joinpoints. However, it decreases the values of WOC and LOC metrics since it merges two separate advices, each has its own signature, into a single advice. In summary, *Merge Advice Bodies* reduces the aspect size in terms of WOC and LOC.
- **Generalize before or after Advice to around Advice:** does not affect DIT and CBC metrics. Nevertheless, it reduces the values of WOC and LOC metrics since it converts two different advices to an around advice with a single signature. In summary, *Generalize before or after Advice to around Advice* decreases the aspect size in terms of WOC and LOC metrics.

Table 3. Classification of AOR based on the Internal Quality Attributes

AOR	DIT	CBC	LCOO	WOC	NOA	LOC
Push Down Inter-type Declaration	-	↓	↓	↓	-	↓
Move Static Introduction	-	↓	↓	↓	-	↓
Eliminating Borrowed Pointcut	-	↓	-	-	↓	↓
Merge Advice Bodies	-	-	-	↓	-	↓
Generalize before or after Advice to around Advice	-	-	-	↓	-	↓
Pull Up Advice	-	-	↓	↓	-	↓
Eliminating Duplicated Pointcut	-	-	-	-	↓	↓
Change Advice kind from Before to After	-	-	-	-	-	-
Change Advice Kind from Around	-	-	↑	↑	-	↑
Extract Method from Advice	-	-	↑	↑	-	↑

- **Change Advice kind from Before to After:** replaces the keyword after in the advice signature by keyword before. Consequently, only the joinpoint that is referenced by the related pointcut is changed. Therefore, this refactoring method does not affect any of the investigated internal quality metrics for the aspect.
- **Pull Up Advice:** does not affect DIT and CBC. However, it decreases the values of LCOO, WOC and LOC metrics of the sub-aspects since it moves an advice from the sub-aspects into the super-aspect. In summary, *Pull Up Advice* makes the sub-aspect more cohesive since and decreases the size of the sub-aspects.
- **Push Down Inter-type Declaration:** does not affect depth of inheritance metric since it doesn't inherit an aspect. However, it decreases the values of CBC, LCOO, WOC and LOC metrics of the super-aspect since it moves an intertype declaration from the super-aspect into the sub-aspect. In summary, *Push Down Inter-type Declaration* decreases the coupling and the size of the super-aspect. Furthermore, it makes the super-aspect more cohesive.
- **Eliminating Borrowed Pointcut:** does not affect cohesion and depth of inheritance metrics. However, the value of CBC metric is decreased since new aspect is created and the existing aspects use the pointcut moved to the new aspect. In addition, the value NOA and LOC metrics of the old aspect are decreased. In summary, *Eliminating Borrowed Pointcut* decreases the coupling and the size of the old aspect.
- **Eliminating Duplicated Pointcut:** does not affect coupling, cohesion and depth of inheritance metrics. However, the values of NOA and LOC metrics of the existing aspects are decreased since new aspect is created and all the duplicated pointcuts in the existing aspects are moved to the new aspect. In summary, *Eliminating Duplicated Pointcut* decreases the size of the source aspect in terms of NOA and LOC metrics.
- **Move Static Introduction:** does not affect DIT metric. However, it decreases the values of CBC, LCOO, WOC and LOC metrics of the aspect since it moves an introduction from an aspect to another. In summary, *Move Static Introduction* decreases the coupling and the size of the aspect; also it makes the aspect more cohesive.

6.2 Effects of AOR on the Software Maintainability

In order to assess the effects of the investigated AOR based on the software maintainability, we mapped the occurred changes in the AO metrics to the maintainability attribute based on [18] in which they found that the metrics DIT, CBC, WOC, NOA and LOC to be inversely proportional to the software maintainability. Then we classified the AOR based on their similar effects on the maintainability.

Table 4. Classification of AOR based on the Software Maintainability

AOR	Maintainability
Merge Advice Bodies	↑
Generalize before or after Advice to around Advice	↑
Pull Up Advice	↑
Push Down Inter-type Declaration	↑
Eliminating Borrowed Pointcut	↑
Eliminating Duplicated Pointcut	↑
Move Static Introduction	↑
Change Advice kind from Before to After	-
Change Advice Kind from Around	↓
Extract Method from Advice	↓

The classification of the investigated AOR based on the software maintainability using the empirical results from all studied software projects is presented in Table 4. In the Table, three different symbols are used: “↑” represents an improvement in the software maintainability, “↓” symbol represents impairment in the maintainability and “-” symbol represents no changes in the maintainability attribute.

From Table 4, it is obvious that seven out of the ten selected AOR can be applied to enhance the software maintainability, while only two of the investigated AOR can impair maintainability when they are applied. However, the results show that “*Change Advice kind from Before to After*” has no impact on software maintainability.

7 Discussion

This section introduces a brief summary of the obtained results from our study, then threats to validity are shown next.

7.1 Summary of the Results

This study provides a number of interesting results which can be observed as follows.

- First, it is observable from Table 3 that “Change Advice Kind from Around” and “Extract Method from Advice” increase the aspect size in terms of WOC and LOC metrics since they introduce new operations. On the other hand, “Merge Advice Bodies”, and “Generalize before or after Advice to around Advice” reduces the source aspect size in terms of WOC and LOC metrics as it merge two advices to form one advice.

- Second, the results support the earlier explanation that “Eliminating Borrowed Pointcut” and “Eliminating Duplicated Pointcut” increases the number of aspects in the system.
- Third, we can notice an interesting observation from Table 4 that is, all the investigated AOR that were applied between different aspects improve the software maintainability. That is as a result of placing the different AO constructs in a more suitable place. On the other hand, the two AOR having negative impact on software maintainability and the one having no impact on maintainability belong to the category called AOR within aspects.
- Fourth, when the objective of refactoring is to enhance the software maintainability, AOR that showed enhancements for the maintainability can be applied and the AOR that impaired software maintainability should be avoided when refactoring to enhance the maintainability.

7.2 Threats to Validity

There are some limitations to the extent to which these results can be generalized. The following are possible reservations: First, there is no universal AOR catalogue like that proposed by Fowler [3] for OO software. Thus we collected the AOR from different sources. Another possible threat is the simplicity of the measures that have been used in this study to assess software maintainability. Finally, there is a lack of the tools for collecting AO metrics, so the data was partially collected by using an available tool [31].

8 Conclusion and Future Work

In this paper, we proposed a classification of AOR based on their measurable effect on (i) internal quality metrics and (ii) software maintainability which is one of the external software quality attributes. The proposed classification can be utilized as guidelines to help software developers decide which AOR can be applied to optimize a software system with respect to the maintainability.

The resulting classification indicates that seven of the out of ten investigated AOR have a positive effect on the software maintainability. On the other hand, two of the selected AOR impair the maintainability. Only one technique from the investigated AO has no effect on the software maintainability.

Additional research directions that can be explored in future work would be classifying the AOR based on other software quality attributes such as reusability and testability. It is also interesting to classify a more extended set of refactoring methods based on software quality attributes to form a large classification catalog.

Acknowledgment

The authors acknowledge the support of King Fahd University of Petroleum and Minerals. This work is done under IN090015.

References

1. Khatchadourian, R.: Aspects of AOP: An Exploration of the Aspect-Oriented Paradigm (2006)
2. Bravo, F.M.: A Logic Meta-Programming Framework for Supporting the Refactoring Process, in Master Thesis. Vrije Universiteit Brussel, Belgium (2003)
3. Fowler, M., Beck, K., Brant, J., Opdyke, W., Roberts, D.: Refactoring: Improving the Design of Existing Code. Addison Wesley, Reading (2000)
4. Filman, R.E., et al.: Aspect-Oriented Software Development. Professional. Addison Wesley Professional, Reading (2004)
5. Kiczales, G.: Aspect-Oriented Programming. In: Aksit, M., Auletta, V. (eds.) ECOOP 1997. LNCS, vol. 1241, Springer, Heidelberg (1997)
6. Laddad, R.: AspectJ in Action – Practical Aspect-Oriented Programming. Manning (2003)
7. Boehm, B., Basili, V.R.: Software defect reduction top 10 list. Computer 34(1), 135–137 (2001)
8. Shahid Nazar Bhatti, J.: Why Quality? ISO 9126 Software Quality Metrics (Functionality) Support by UML Suite. ACM SIGSOFT Software Engineering Notes, 30(2), (2005)
9. Coleman, D., Ash, D., Lowther, B.: Using Metrics to Evaluate Software System Maintainability. Computing Practices (1994)
10. Tian, Y., Chen, C., Zhang, C.: AODE for Source Code Metrics for Improved Software Maintainability. In: Fourth International Conference on Semantics, Knowledge and Grid (2008)
11. Zakaria, A., Hosny, H.: Metrics for Aspect-Oriented Software Design. In: Proceedings of Workshop on Aspect-Oriented Modeling, International Conference on Aspect-Oriented Software Development (2003)
12. Kvale, A.A., Li, J., Conradi, R.: A Case Study on Building COTS-Based System Using Aspect-Oriented Programming. In: SAC 2005, Santa Fe, New Mexico, USA (2005)
13. Madeyski, L., Ia, L.S.: Impact of aspect-oriented programming on software development efficiency and design quality: an empirical study. IET Software Journal 1(5), 180–187 (2007)
14. Tonella, P., Ceccato, M.: Refactoring the Aspectizable Interfaces: An Empirical Assessment. IEEE Transactions on Software Engineering 31(10), 819–832 (2005)
15. Kulesza, U., Sant'Anna, C., Garcia, A., Coelho, R., Staa, A., Lucena, C.: Quantifying the Effects of Aspect-Oriented Programming: A Maintenance Study. In: Proceedings of the 22nd IEEE International Conference on Software Maintenance (2006)
16. Sant'Anna, C., Lobato, C., Kulesza, U.: On the modularity assessment of aspect-oriented multiagent architectures: a quantitative study. Int. J. Agent-Oriented Software Engineering 2(1), 34–61 (2008)
17. Fenton, N.E., Pfleeger, S.L.: Software Metrics- A Rigorous and Practical Approach. PWS Publishing Company (1997)
18. Sant'Anna, C., Garcia, A., Chavez, C., Lucena, C., von Staa, A.: On the Reuse and Maintenance of Aspect-Oriented Software: An Assessment Framework. In: Proceedings of the Brazilian Symposium on Software Engineering (2003), PUC-RioInf.MCC26/03
19. Basili, V.R., Briand, L.C., Melo, W.L.: A Validation of Object-Oriented Design Metrics as Quality Indicators. IEEE Transactions on software engineering 22(10) (1996)
20. Chidamber, S.a.C.K.: Towards a Metrics Suite for Object-Oriented Design. In: Proceedings of the Conference on Object Oriented Programming Systems Languages, and Applications OOPSLA 1991 (1991)

21. Chidamber, S.R., Kemerer, C.F.: A Metrics Suite for Object Oriented Design. *IEEE Transactions on Software Engineering* 20, 476–493 (1994)
22. Ceccato, M., Tonella, P.: Measuring the Effects of Software Aspectization. In: *Proceedings of the 1st Workshop on Aspect Reverse Engineering* (2004)
23. Tsang, S.L., Clarke, S., Baniassad, E.: An Evaluation of Aspect-Oriented Programming for Java-based Real-time Systems Development. In: *Proceedings of the Seventh IEEE International Symposium on Object-Oriented Real-Time Distributed Computing ISORC 2004* (2004)
24. Greenwood, P., Bartolomei, T., Figueiredo, E., Dosea, M., Garcia, A., Cacho, N., Sant'Anna, C., Soares, S., Borba, P., Kulesza, U., Rashid, A.: On the impact of aspectual decompositions on design stability: An empirical study. In: *Batani, M. (ed.) ECOOP 2007. LNCS, vol. 4609, pp. 176–200. Springer, Heidelberg* (2007)
25. Monteiro, M.J.T.P.: Refactorings to Evolve Object-Oriented Systems with Aspect-Oriented Concepts, in Ph.D. Thesis. University of Minho, Portugal (2005)
26. Monteiro, M.P.: Catalogue of Refactorings for AspectJ., Technical Report Um-Di-Gecsd-200402 (2004)
27. Rura, S., Refactoring Aspect-Oriented Software. In: *Undergraduate Thesis in Computer Science. Williamstown, Massachusetts, Williams College* (2003)
28. Feremans, L.: Aspect-Oriented Refactoring, in MS Thesis. Vrije Universiteit Brussel (2005)
29. Srivisut, K., Muenchaisri, P.: Defining and Detecting Bad Smells of Aspect-Oriented Software. In: *Proceedings of the 31st Annual International Computer Software and Applications Conference COMPSAC 2007* (2007)
30. Al-Jamimi, H.A.: Classification of Refactoring Methods for Aspect-Oriented Programming based on Software Quality Attributes. In: *ICS Department, King Fahd University of Petroleum & Minerals, Saudi Arabia* (2010)
31. Figueiredo, E., Garcia, A., Lucena, C.: AJATO: An AspectJ Assessment Tool. In: *European Conference on Object-Oriented Programming ECOOP Demo, France* (2006)
32. Jhotdraw project, <http://sourceforge.net/projects/ajhotdraw/>
33. Tetris in AspectJ project,
<http://www.guzzzt.com/coding/aspectttetris.shtml>
34. AspectJ Development Tools, <http://www.eclipse.org/ajdt/>
35. AspectJ Exception FrameWork proj.,
<http://sourceforge.net/projects/ajefw/>

On the Modelling of Adaptive Hypermedia Systems Using Agents for Courses with the Competency Approach

Jose Sergio Magdaleno-Palencia¹, Mario Garcia-Valdez¹,
Manuel Castanon-Puga², and Luis Alfonso Gaxiola-Vega

¹ Tijuana Technological Institute, Mexico

² Baja California Autonomous University, Mexico

{sergio.magdaleno,puga,lgvega}@uabc.edu.mx,
mario@tectijuana.edu.mx

Abstract. This work is motivated by the need to propose a model for the study of personalized hypermedia systems with a competency approach. We use computer agents, to construct a student model with emphasis on his learning style, using different multiple intelligences as well the competency model to adapt the course material to the student's needs. The system will use agent's technology as well artificial intelligence techniques. The systems will be used in a university setting; the primary goal is to help each student learn and thus reach a suitable competency level. In this paper the model's requirements are presented.

Keywords: Adaptive Hypermedia System, Agents, Competency model, Learning style, Multiple intelligences.

1 Introduction

Each person learn in a different way, each one processes and perceives information differently; hence the importance to create a system to adapt these individual characteristics for each one. Studies shows that when presented information to the people according to their learning style, they learn better [1].

The primary goal is to create a system that takes a novel approach to using personalized educational hypermedia components using students profiling and the competency model. The system will be applied in courses with the competency approach in higher education.

It is important to remember that one of the UNESCO recommendations in education is to promote lifelong learning. For this purpose the following topics are necessary to create the system.

1.1 Adaptive Hypermedia System

Adaptive hypermedia systems have been used as software tools in the teaching of courses; they allow for adaptation according to the users learning style by making the necessary adjustments in the way the course material is presented [2], that's why it

is called a personalized hypermedia system. The classical architecture for an adaptive hypermedia system includes the user, adapter and domain models; each one of them to know the student's profiling, the process to adapt the domain according to the student's needs and the domain itself.

1.2 User Model

The user model includes for their profiling the learning style and multiple intelligences as well the evaluation of the course material during the process.

1.2.1 Learning Style

Each person learns in a different way, each one processes and perceives information differently, thus according to their own learning style. A work done by Neil Fleming at Lincoln University was launched in 1987, through this work the word VARK was used to stand for Visual, Aural, Read/write, and Kinesthetic sensory modalities that are used for learning information [3].

Alonso, Gallego and Honey in their research indicate that students learn better when they know their predominant learning style [1].

1.2.2 Multiple Intelligences

Dr. Howard Gardner developed the theory of multiple intelligences in 1983; he suggested that the traditional notion of intelligence, based on I.Q. testing, is far too limited. Instead, Dr. Gardner proposes eight different intelligences to account for a broader range of human potential. These intelligences are: linguistic, logical-mathematical, spatial, bodily-kinesthetic, musical, interpersonal, intrapersonal and naturalist[4]

1.3 Competency Model

The educational model of competency is one of the UNESCO recommendations to promote lifelong learning and competency building, as a result of a globalized world of communications. Hence four pillars of education (learning to know, learning to do, learning to live together and learning to be) are promoted by UNESCO worldwide [5]. This model integrates the types of known learning outcomes known, the theoretical domain declarative or relating to knowledge, procedural skills or know-how relating to (run or do something, using techniques, methods, skills and strategies [6]) and the affective or attitudinal, knowing being, which are three of the four pillars.

1.3.1 Competency Evaluation

The project will be using the competency approach, with competency standards and professional requirements derived from manufacturing and service sector; it forms the learner through a teaching methodology that emphasizes the know-how and uses an organization and infrastructure similar to the area where these skills will be used [7]. Hence we will be using the competency evaluation according to the work done by Guzman 2009.

1.4 Multi-agent Model

An agent is “A computer program, or part of a program, that can be consider to act autonomously and that represents an individual, organization, nation-state, or other social actor.”, [8]

An agent is situated in some environment, within which it can act independently and flexibly to meet its main objective. The agent will receive data from its environment and must choose an appropriate action. According to the authors Wooldridge and Jennings, agents have certain properties such as: autonomy, sociability, reactivity, pro-activity, intelligence, rationality, consistency, and adaptability,[9]. “A multi-agent system (MAS) can therefore be defined as a collection of, possibly heterogeneous, computational entities, having their own problem solving capabilities and which are able to interact among them in order to reach an overall goal”, [9]. The learning process is a complex system involving different types of very complex agents, for instance a community of students each one with their own personality, learning style, background, motivation, culture etc. Multi agents systems can be very powerful tools to model this kind of systems, having the capacity to simulate and recreate this behavior. The MAS proposed system would be used to learn about the learning process and can be used to design future adaptive learning environments.

2 Adaptive Hypermedia System Using Agents with the Competency Approach

Figure 1 shows the proposed system architecture for the personalized Adaptive hypermedia system using agents for courses with the competency approach (PAHSUACCA).

We proposed a model based on three agents: User Profile Agent responsible of the precise diagnosis of the student’s model which includes his learning style, type of intelligence and background knowledge. An Adapter Agent responsible for the selection of the appropriate learning activities in accordance with the competency approach selected by the Competency Model Agent. All of them integrated in their environment.

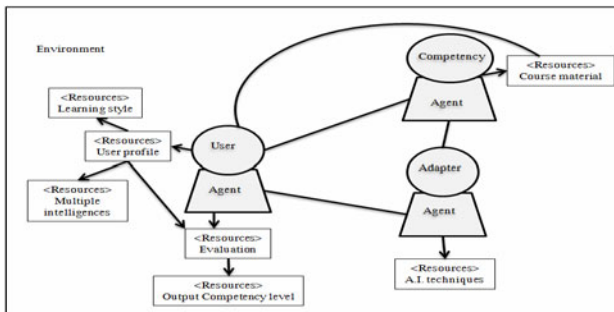


Fig. 1. PAHSUACCA Model

This methodology requires information from the user's background, such as learning style and type of intelligence; so the system can create the user profile. Hence the User Agent uses the learning style and multiple intelligences resources to create the User profile and also use the evaluation done with the use of the system. The course material is tailored with the competency approach, the Competency Agent validate and follows the competency model's strategies and the Agent adapter will use Artificial Intelligence (A.I.) techniques to personalize the material according to the user's profile and the results of the evaluation for each topic of the course and present it to the user in a way that will be easier to understand. This process will be in a loop until the user can reach the competency level required. The system evaluation will use the competency evaluation approach [7]. For this project each component will be handled by an agent. All the agents will be interacting with the environment.

The system will use Bloom's Taxonomy to help adaptation to the student's learning style; it considers the cognitive domain, in respect to any area of knowledge. The types of learning's domains suggested by Bloom are: cognitive (knowledge understanding, application, analysis, synthesis and evaluation), affective and psychomotor [10].

It is important to know that the adaptation should be according the domain, the profiling and the student's performance that will be measure by the competency evaluation.

3 Case Study

One of the commitments' of the General Administration of Higher Education Technology (DGEST) in Mexico, is to seek comparability, compatibility and competitiveness of the plans and study programs in a national and international context [11]. Hence the case study will be implemented in Tijuana Technological Institute, Mexico, with primary goal is to help each student learn and thus reach a suitable competency level thus according the UNESCO recommendation.

3.1 Applications of Adaptive Hypermedia in Education

Advances on information systems and technology shows that it is very important to focus on developing competency-based learning models [12]. Till this day the development of Adaptive Hypermedia Systems (SHA), has been instrumental in helping students achieve meaningful learning. It has also been developed due to the impact of media in the learning style of young people, new needs of life imposed by globalization, advances in modern psychology, changes in the aims of education [5], [13].

New e-courses tend to be oriented to support the activities according to the constructivism theory [14], hypermedia systems have become popular in the last years as tools for student' learning [15]. Some projects have been carried out for competency-based training standards, instructional technology education [16],[17], and competency-based systems for recommending study materials from the Web to

learners (CBSR). It explores the benefits of a competency model for an improved pedagogical approach to e-learning [18] and also Adaptive Learning Objects Sequencing for Competence-Based Learning [19].

The competencies are distinguished by their types, the most common are conceived as: basic, generic and specific. The components are broken down into three levels such as: general competency, the unit of competency and elements of competency [20].

Some systems adapt A.I. techniques for the recommendations [21], but they do not use the competency model for the pedagogical material. Also some hybrid system architecture for learning objects [22] had been done.

With the implementation of this research, we pretend to know the following: Which are the main characteristics of adaptability that the educational resources should have with a competency approach?, which are the main characteristics of adaptability that the evaluation mechanism should have for the student's knowledge level based on the competency model?, which Artificial Intelligence techniques are for diagnosing and evaluating the student's knowledge in the competency approach?

3.2 Design and Implementation

For the implementation of the PAHSUACCA Model we are going to use the following: for the user (client) CSS, XHTML 5.0 and Javascript; for testing the systems we are going to use Firefox from Mozilla as well Internet Explorer from Microsoft; for the server application CherryPy 3.0; to connect the management system database we are going to use sqlite3 and to program all the other system's components the Python language.

4 Conclusions

According to the UNESCO recommendation of to promote lifelong learning and competencies building [5], the adoption of the model by the education systems [23] and one of the commitments' of the General Administration of Higher Education Technology (DGEST) in Mexico, to seek comparability, compatibility and competitiveness of the plans and study programs in a national and international context [11].

The proposed methodology allows the creation of personalized system with the competency approach, using a precise diagnosis of students learning with their profile and adapts the best strategy using A.I. techniques to obtain better result for each student, and improves the learning in higher education institutions.

Moreover, the technology of multi-agent systems has created opportunities for improved, AHS, to incorporate the features characteristic of an agent, such autonomy, sociability, reactivity, pro-activity, intelligence, rationality, consistency, and adaptability [9].

References

1. Alonso, C., Gallego, D., Honey, P.: *Estilos de aprendizaje, los procedimientos de diagnóstico y mejora*. 6 edición ed, Bilbao, Ediciones Mensajero (2001)
2. Llamosa, R., et al.: *Sistema hipermedia adaptativo para la enseñanza de los conceptos básicos de la programación orientada a objetos*. XI CIES (November 18, 2003)
3. Fleming, N.D., Mills, C.: *Mills, Not Another Inventory, Rather a Catalyst for Reflection*. *To Improve the Academy* 11 (1992)
4. Armstrong, T.: *7 Clases de inteligencia. Identifique y desarrolle sus inteligencias multiples*, Mexico: Editorial Diana (2002)
5. Delors, J.: *La educación encierra un tesoro*. UNESCO, Mexico (1997)
6. Valls, E.: *Los procedimientos: aprendizaje, enseñanza y evaluación*. In: Horsori, S.L. (ed.), p. 184 (1993)
7. Guzmán, J.: *¿Cómo evaluar en competencias educativas? Diseñe instrumentos y métodos psicopedagógicos eficaces*, Bogotá, Colombia: Psicom editores (2009)
8. Gilbert, N.: *Agent-Based Models*. Sage Publications, London (2007)
9. Jennings, N.R.: *Agent-oriented software engineering*. In: Imam, I., Kodratoff, Y., El-Dessouki, A., Ali, M. (eds.) IEA/AIE 1999. LNCS (LNAI), vol. 1611, pp. 4–10. Springer, Heidelberg (1999)
10. Fowler, B.: *La taxonomía de Bloom y el pensamiento crítico*, (2002), <http://www.eduteka.org/profeinvitad.php3?ProfInvID=0014> (cited, February 11, 2011)
11. García, C.: *Eventos de Diseño Curricular Basado en Competencias Profesionales*, (2010), <http://www.dgest.gob.mx/docencia/eventos-de-diseno-curricular-basado-en-competencias-profesionales> (cited November 1, 2011)
12. Voorhees, R.: *Competency-Based Learning Models: A Necessary Future*. In: *New Directions For Institutional Research*, John Wiley & Sons, Inc., Chichester (2001)
13. Frade, L.: *Desarrollo de competencias en educación: desde preescolar hasta el bachillerato*, ed. a edición, inteligencia educativa, México, D.F (2009)
14. Bruner, J.: *Toward a Theory of Instruction*. Harvard University Press, USA (1996)
15. Brusilovsky, P.: *Methods and techniques of adaptive hypermedia*. *User Modeling and User-Adapted Interaction* 6(2), 87–129 (1996)
16. Spector, J., Klein, J., Reiser, R., Sims, R., Grabowski, B.: *Competencies and Standards for Instructional Design and Educational Technology* (2006), <http://www.ibstpi.org/competencies.htm> (cited December 20, 2010)
17. Carro, R.: *Applications of Adaptive Hypermedia in Education*, in *Computers and Education*. In: *Towards Educational Change and Innovation*, pp. 1–12. Springer, Heidelberg (2008)
18. Nitchot, A., Gilbert, L., Wills, G.B.: *A Competence-based system for recommending study materials from the Web (CBSR)*. Learning Soc. Lab. Univ of Southampton, UK (2010)
19. Karampiperis, P., Sampson, D.: *Adaptive Learning Objects Sequencing for Competence-Based Learning*. In: *Sixth IEEE International Conference on Advanced Learning Technologies, ICALT 2006* (2006)

20. Saluja, S.: La capacitación basada en competencias en el Reino Unido”. En Arguelles A (comp.). *Competencia Laboral y Educación Basada en Normas de Competencia*. Limusa-Noriega Editores, México (1996)
21. Garcia, M.: Aprendizaje colaborativo basado en recursos adaptativos, in Departamento de Ciencias Químicas e Ingeniería. Universidad Autónoma de BajaCalifornia: Tijuana (2008)
22. Garcia, M., Parra, B.: A Hybrid Recommender System Architecture for Learning Objects. In: Castillo, O., Pedrycz, W., Kacprzyk, J. (eds.) *Evolutionary Design of Intelligent Systems in Modeling, Simulation and Control*. Studies in Computational Intelligence, vol. 257, pp. 205–211. Springer, Heidelberg (2009)
23. Climent, J.: Sesgos comunes en la educación y la capacitación basadas en estándares de competencia. *Revista Electrónica de Investigación Educativa*, 12(2), (cited December 20, 2010)

Toward a Methodological Knowledge for Service-Oriented Development Based on OPEN Meta-Model

Mahdi Fahmideh¹, Fereidoon Shams¹, and Pooyan Jamshidi²

¹ Automated Software Engineering Research Group,
ECE Faculty, SB University GC, Tehran, Iran
{m_fahmideh, f_shams}@sbu.ac.ir

² Lero - The Irish Software Engineering Research Centre,
School of Computing, Dublin City University, Dublin, Ireland
pooyan.jamshidi@computing.dcu.ie

Abstract. Situational method engineering uses a repository of reusable method fragments that are derived from existing software development methodologies and industrial best practices to simplify the construction of any project-specific software development methodology aligned with specific characteristics of a project at hand. In this respect, OPEN is a well-established, standardized and popular approach for situational method engineering. It has a large repository of reusable method fragments called OPF that method engineers can select and assemble them according to the requirements of a project to construct a new project-specific software development methodology. In this position paper, we present the basic concepts and foundations of OPEN and argue for an urgent need for new extensions to OPEN and its repository in support of service-oriented software development practices.

Keywords: OPEN Process Framework, OPF Repository, OPEN Meta-Model, Situational Method Engineering, Method Fragments, Service-Oriented Software Development.

1 Introduction

It has proven untenable that there is no universal Software Development Methodology (SDM) appropriate for all situations [1,2,3]. This consensus is due to the situation factors of projects at hand such as organizational maturity and culture, people skills, commercial and development strategies, business constraints, and tools issues. Software development organizations need to develop their own project-specific SDM for their software projects. In this respect, Method Engineering (ME) is an approach in which a project-specific SDM is constructed. The most well-known offered sub-set of ME for tailoring SDMs is called Situational Method Engineering (SME) [4,5,6] wherein a project-specific SDM is constructed from the assembly of a number of reusable Method Fragments [7] or Method Chunks [8] that are stored in a Repository or Method-Base [9,10]. Indeed the repository of method fragments is a Methodological Knowledge-Base that stores the knowledge about what and how to develop a software system [11]. Each method fragment bears a piece of knowledge

for software development that is characterized by a name and an intention specifying the goal of the method fragment. More specifically, a method fragment can be thought of as a couple of two interrelated parts: product model and process model. The product part of a method defines a set of concepts and relationships between these concepts. In contrast, the process part describes how to construct the corresponding product part. Fig. 1 depicts a typical method fragment. This method fragment aims to provide a use-case model as a solution to resolve a problem description. The product part represents the required products and the process part provides suitable guidelines to make a use-case model.

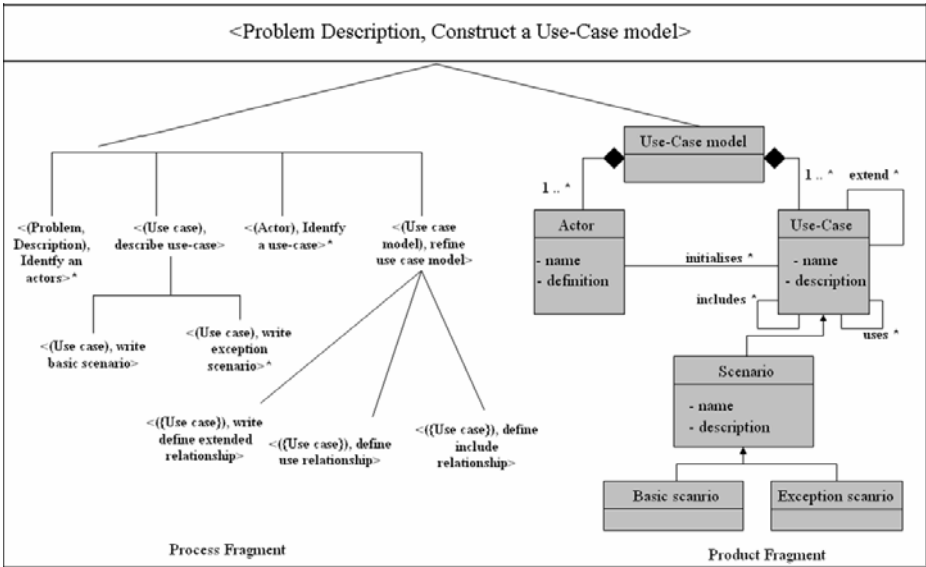


Fig. 1. A method fragment (adopted from [11])

The Object-oriented Process, Environment, and Notation (OPEN) or Open Process Framework (OPF) [12] is a framework that is highly compatible with the ideas of SME approach. OPEN has a large number of method fragments stored in a repository, called OPF. Indeed the OPF repository is a methodological knowledge-base. In OPEN, methodological knowledge is represented in the OPEN meta-model format as well as method fragments. A method engineer can select and assemble method fragments and construct a project-specific SDM based on the unique set of characteristics of the project at hand. However, OPF contains method fragments mainly intended for Object-Oriented (OO) software development.

Given that newer approaches to software development have emerged that are currently practiced widely, there is an urgent need to enhance the OPF repository with new method fragments in support of these new approaches to software development [10]. One of these new and popular approaches is Service-Oriented Computing (SOC) that has been proposed in the last few years and is getting huge interest from software engineering researchers and practitioners. As stated in [13,14,15], SOC is a new

computing approach that utilizes software services as the fundamental building blocks to facilitate the development of rapid, low-cost and easy composition of loosely-coupled fully distributed systems such as Clouds. A strand of inclination in the SOC field is the Service-Oriented (SO) software development subfield that grapples with development of SO software systems [15]. It should be noted it is needed to apply SME idea approach to the SO software development context [16,17,18] because (1) the development of SO software are increasingly decentralized, (2) SO software is composed dynamically out of parts that are developed and operated by independent parties, (3) changes in requirements ask for continuous SO software adaptation and evolution, and that (4) the infrastructures on which SO applications run are fully distributed. This situation fosters the need for adoption of SME approaches in order to satisfy the various and changing requirements of SO software development.

Although OPEN has a large repository in support of construction of various types of software such as OO and Component-Based (CB) systems, it lacks any support for the development of SO systems. It is thus reasonable for the OPF repository to be extended to provide support for SO in addition to its current support for OO and CB software development. Therefore, in this paper we take a deep look at the foundations and basic constituents of OPEN and argue for extending OPF with method fragments in support of SO software development.

We have organized the rest of the paper as follows. Section 2 presents a brief overview of SME and OPEN. Section 3 argues in favor of adding SO extensions to the OPF repository. Section 4 concludes the paper with proposals for future work.

2 Overview

2.1 Situational Method Engineering

The prevalent belief that no single SDM can be applicable to all situations is the main reason for the emergence of ME [19]. Each software project has different characteristics so that a SDM tailoring should be adopted before starting the project. The ME approach was first introduced by Kumar [20] as a software engineering discipline aimed at constructing project-specific SDMs to meet organizational characteristics and projects situations. Brinkkemper [21] elaborated the definition of ME later as: “the engineering discipline to design, construct, and adapt methods, techniques and tools for the development of information systems”. The most well known subset of ME is SME that is concerned with the construction, adaptation and enhancement of suitable SDM for the project at hand instead of looking for universal or widely applicable ones [21]. Based on the SME approach, a SDM is constructed from a number of encapsulated method fragments that have been already stored in a repository. The method fragments are the atomic elements of any SDM that a method engineer extracts from existing SDMs or from industrial best practices [22,23]. Method fragments are selected in such way to satisfy target SDM’s requirements.

To realize SME, researchers have proposed many approaches [19]. Typically, the steps below are followed to construct a project-specific SDM:

1. Method engineer elicits and specifies the requirements of the target SDM based on the characteristics of the project at hand.

2. Then, he/she selects a number of most relevant method fragments from the repository based on a number of factors highly specific to the particular software development organization and particular situation of the project.
3. Method fragments are assembled to construct a full project-specific SDM.
4. To ensure high quality of the constructed SDM, a list of assessment criteria such as the ones proposed by Brinkkemper is used [24].

SME has been extensively used for OO software development [25]. One instance of the SME approach that is highly compatible and fits well with the above steps is extensively used for the development of a wide range of software projects, especially in the OO context is called OPEN [12]. OPEN defines a process meta-model that allows the elements of the OPEN, i.e method fragments, to be represented and reused. A large number of method fragments, stored in a single repository, called OPF repository, facilitates the instantiation of any project-specific SDM from the OPEN. The method fragments are reusable building blocks that can be adopted in more than one SDM construction effort [26]. Predefined rules and construction guidelines assist method engineers to select from repository and to assemble them. The instantiated SDM is mainly a new configuration of the OPEN. Successful industrial use of OPEN demonstrates its viability to software development [27]. This utilization in real world practices was the main reason we were motivated to extend OEPN with SO support.

2.2 OPEN Process Framework

OPEN or OPF is the oldest established SDM introduced in 1996 in an effort to integrate four SDMs namely, MOSES, SOMA, Synthesis and Firesmith [12,28]. OPEN is known as a popular SDM with full iterative-incremental lifecycle and process-focused SDM that is recently updated to become conformant with ISO/IEC 24744 [29]. It is mainly intended for use in either the development of a wide range of software systems or the construction of a wide-range of project-specific SDMs. OPEN is maintained by a not-for-profit Consortium consisting of an international group of methodologists, academics and CASE tool vendors [30]. As shown in Fig.2, OPEN contains an underpinning process meta-model, a single rich repository of method fragments, supportive tools, and usage guidelines that explain how method engineers can deploy the method fragments.

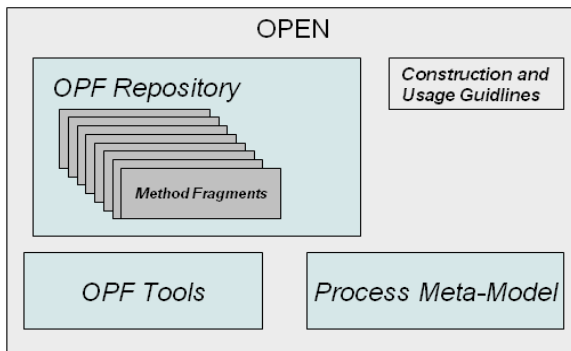


Fig. 2. The main elements of the OPEN Process Framework (adopted from [12])

The OPEN's process meta-model provide a clear way to formally represent any method fragments e.g. process models, phases, activities, tasks, techniques, work products and roles. It is imperative that method fragments conform to the OPEN meta-model standard. This implies that new method fragments to be added to the repository must be conformant to this meta-model as well. The OPEN meta-model as shown in Fig.3 contains five core classes of method fragments as defined by Firesmith and Henderson-Sellers [12,30]:

1. **Work Unit:** Operations should be performed by Producer(s) or tools to develop required Work Products. Work Units based on their granularities are categorized in three levels of abstractions:
 - **Activity:** Some refer to Activity as software engineering discipline too. An activity is a coarse-grain type of typical Work Unit consisting of a cohesive collection of Tasks that produce a related set of Work Products. In other words, an Activity includes a group of relevant Tasks.
 - **Task:** A Task is a fine-grain type of Work Unit consisting of a cohesive collection of steps that produce Work Product(s).
 - **Technique:** An explicit procedure(s) that explains how a Task should be performed is called a Technique.
2. **Work Product:** Work Product is any significant produced artifact such as diagram, graphical and textual description, or program that is produced during software development.
3. **Producer:** Person(s) or tools that develop expected Work Products are a kind of Producer.
4. **Language:** Language represents the produced artifacts using a modeling language such as Unified Modeling Language (UML) [31] or any implementation language.
5. **Stage:** Stage is used for defining the overall macro-scale and time-box on a set of cohesive Work Units during the enactment of an instantiated OPEN. The instantiated process is structured temporally using the Stage concept element.

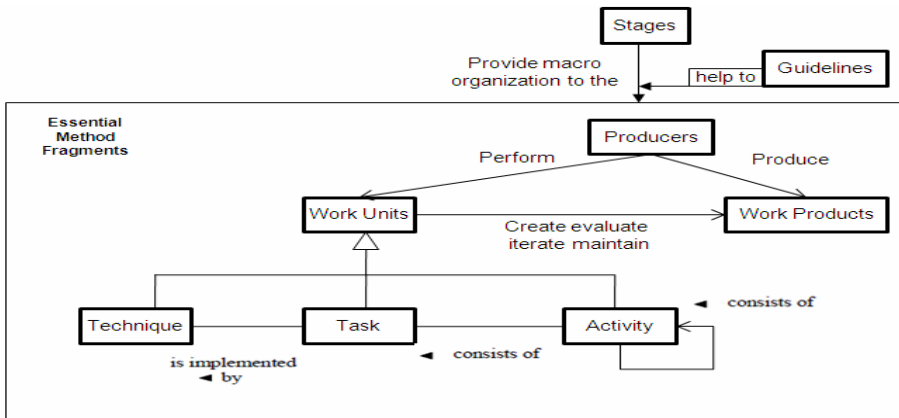


Fig. 3. Constituents of the OPEN Process Meta-Model [12]

In addition to the process meta-model, OPEN contains a large number of method fragments at different levels of granularity (Activities, Tasks and Techniques) stored in a repository as it is shown in Fig.3. The process meta-model and repository of method fragments provide the underpinning and scaffolding context for situational method engineering. The OPF repository provides reusable method fragments as well as well-known and traditional activities for the construction of project-specific SDMs that are mainly intended for OO software development [12,30]. For instance, there are many tasks and techniques for Requirements Engineering such as requirements elicitation, use-case modeling, use-case specification, and prototyping.

There are many other approaches to development of software projects other than OO such as Component-Based Software Engineering (CBSE) or Component-Based Development (CBD). Although there is a lot of commonality between the traditional OO software development and other approaches of software development, the latter differs slightly from OO software development. On the other hand, we know that the most critical pre-requisite for construction of SDM is a repository that should be consistent with paradigmatic approaches [25,32]. Therefore, it is necessary that new method fragments are added to the OPF repository in support of new approaches of software development or technologies. Fortunately, the addition of new method fragments does not require any modifications to the underpinning OPF meta-model that has been standardized [25].

3 Discussion

Over the past years, several researchers have attempted to provide methodological knowledge to the OPF repository in support of various software development approaches. Henderson-Sellers et al have carried out the most sound and significant extension to the OPF repository. They have added many supportive method fragments to facilitate situational SDM construction in different approaches of software development such as:

- Extension Support for CBD [33].
- Extension Support for Web-Based Software Development [34,35].
- Extension Support for Aspect-Oriented Programming (AOP) [32].
- Extension Security-Related method fragments for the OPF [25].
- Several additional extensions in support of organizational transition and usage-centered design [36,37,38].

Therefore, OPF has matured with the introduction of method fragments for various approaches of software development. Recently, SOC has become more and more popular [13,14,15]. SOC has many favorable attributes including agility, reusability and standardization in a technical and in a business-oriented sense. In this respect, SO software development has received wide acceptance among practitioners and academia. Mainly, SO software development is considered as the next step towards resolving the deficiencies of CBD approach such as 1) the lack of standard interface that has been very difficult for software developers for component interoperability [15] and 2) ignorance of security issues of software components [39]. In the SO context, standard interfaces provide greater interoperability between service providers

and consumers. Specifically SO is considered as an evolution of CDB software development [40]. In the context of SO, services are defined as reusable platform-independent building blocks of the system (or software). The standard interfaces provide greater interoperability between service providers and consumers and simplify the development of loosely coupled distributed systems [13]. This approach of software development is known as Service-Oriented Architecture (SOA), Service-Oriented System Development, or Service-Oriented Software Engineering (SOSE). Since the SO approach can significantly improve the way software systems are developed, there has been an increase in the tendency for developing SO systems [15].

SO software development resembles the traditional waterfall process model and activities such as project planning, use-case modeling, OO Analysis and Design, Implementation and Test. However, many research publications and empirical evidences [15,41,42] report that the development of SO systems is different from the traditional software development. The SO software development has more challenges than traditional software development. Typically, an SO software development constituents activities such as service governance, service identification, specification and realization, service discovery and composition and service monitoring [14,43]. It is increasingly being recognized that modifications to traditional process models to suite the SO development, and introducing new software engineering activities and skills other than traditional activities for SO systems, are required. In addition, the development of SO systems need to apply SME approach to the SO software development in which a project specific SDM should be tailored specifically to meet the requirements of such projects. We strongly believe that there is a need to provide a repository we call it a Methodological Knowledge-Base that is extracted from successful experiences and best practices of SO development, for exchanging methodological knowledge among practitioners and software development organizations. In this context, similar endeavors have been accommodated in the area of requirements engineering [44], business interoperability [45] and method engineering [46]. For instance, the [44] proposed a set of classified and formalized patterns stored in a repository. The patterns represent recurrent problems and solutions during requirements engineering that could be adopted by developers.

An essential need to have such methodological knowledge so that it would be as much as useful is its representation. The SO methodological knowledge should be well-documented and maintained in a well-structured format so that one can easily understand and utilize it in real projects. Integrating different kinds of SO methodological knowledge in a common repository, such as OPF method fragments, has the following benefits:

- **Documentation:** Provides software development organizations with well-documented and well-structured useful knowledge of SO development.
- **Reusability:** The repository can be very useful to construct SO SDMs by assembling existing fragments or to adapt existing SDMs by adding specific fragments.
- **Continuous Evolution:** Getting feedbacks from practitioners, analyzing the feedbacks, and keeping methodological knowledge alive and up to date. Moreover, it is open and allows new practitioners to contribute new method fragments into repository.

- Share knowledge:** To achieve this aim, OPEN is a good candidate because OPEN provides a standard meta-model for representation of methodological knowledge via autonomous and coherent method fragments. Moreover, OPEN provides good support from various approaches of software development and ideas of SME. The methodological knowledge provides support to software development organizations to help them construct project-specific SDM and share knowledge of developing SO systems with other practitioners.

In spite of full support of OPEN for various software development approaches, we have identified a deficiency in the current OPF. To the best of our knowledge, there is no support and similar reported research into defining specific method fragments for SO development. We advertise in support of methodological knowledge for SO development by providing a set of method fragments. These fragments are stored in the OPF repository alongside pre-existing method fragments. The applicability of the extended OPF is that any method engineer can construct a new SDM by selecting from existing method fragments for traditional activities and selecting from SO specific method fragments for SO development.

To achieve a methodological knowledge for development of SO systems in the format of OPEN method fragments, the source in which knowledge is extracted from is the main prerequisite. As stated in [47], one way to construct new method fragments is to utilize the existing SDMs as a source called existing method re-engineering. In this approach, a SDM is fully decomposed into a number of method fragments ready to be stored in the repository. To realize this approach, we have developed a manual procedure for identifying reusable method fragments from SDMs [48]. The procedure gets a SDM and proposes several steps to the method engineer to decompose SDM into a set of method fragments. The constructed method fragments are represented in the OPEN meta-model standard (as mentioned in Section 2.2) which is deemed to enhance the OPF repository so that they can be immediately imported into OPF tools supports. Indeed, the constructed method fragments will be methodological knowledge in SO development approach. As shown in Fig 4, the new method fragments are placed along with other existing method fragments as well.

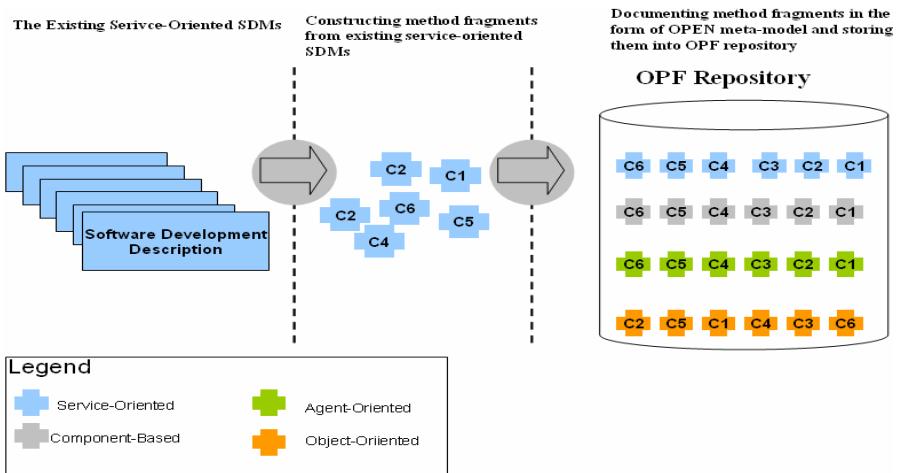


Fig. 4. The overall process of constructing new SO specific method fragments

4 Future Work

We have started a thorough study of the fundamental issues in SO development, specifically in the current prominent SO SDMs, and have short-listed ten candidates. The candidates are IBM SOMA 2008 [43], SUN SOA Repeatable Quality (RQ) [49], CBDI-SAE Process [50], MSOAM [51], IBM RUP for SOA [52], Methodology by Papazoglou [14], IBM SOAD [53], SOUP [54], Steve Jones' Service Architectures [55] and Service-Oriented Architecture Framework [56]. These SDMs prescribe successive systematic activities in order to fulfill SO issues. These SDMs have been selected because of their empirical evidence, higher rate of citations, more accessible resources, and better documentations. In future, we intend to derive the commonalities between these SDMs and propose a set of method fragments. The method fragments convey from OPEN meta-model so they can easily be added to OPF repository. We envisage three levels of granularities for these specific SO method fragments:

- **Activity:** Some of the existing method fragments for Activities in the OPF would be enhanced by the incorporation of SO ideas specific to SO Task method fragments.
- **Tasks:** New SO Task method fragments.
- **Techniques:** For each Task method fragment, a number of supportive Techniques will be provided.

Therefore, future research offers the new SO specific method fragments for OPF repository.

References

- [1] Cockburn, A.: Selecting a project's methodology. *IEEE Software* 17(4), 64–71 (2000)
- [2] Kumar, K., Welke, R.J.: Methodology engineering: a proposal for situation-specific methodology construction. In: Cotterman, W.W., Senn, J.A. (eds.) *Challenges and Strategies for Research in Systems Development*, pp. 257–269. J. Wiley, Chichester (1992)
- [3] Slooten, K., Brinkkemper, S.: A method engineering approach to information systems development. In: Prakash, N., Rolland, C., Pernici, B. (eds.) *Information Systems Development Process Procs. IFIP WG8.1*, Elsevier Science Publishers B.V, North-Holland (1993)
- [4] Ralyte, J.: Towards situational methods for information systems development: engineering reusable method chunks. In: Ralyte, J. (ed.) *Procs. 13 th Int. Conf. on Information Systems Development. Advances in Theory, Practice and Education (Vilnius, Lithuania)*, pp. 271–282 (2004)
- [5] Rolland, C., Prakash, N., Benjamin, A.: A Multi-Model View of Process Modeling. *Requirements Engineering Journal* 4(4), 69–187 (1999)

- [6] Van Slooten, K., Hodes, B.: Characterizing IS development projects. In: IFIP TC8 Working Conference on Method Engineering: Principles of method construction and tool support, London (1996)
- [7] Henderson-Sellers, B., Gonzalez-Perez, C., Ralyté, J.: Comparison of Method Chunks and Method Fragments for Situational Method Engineering. In: Proceedings 19th Australian Software Engineering Conference. ASWEC2008, pp. 479–488. IEEE Computer Society Press, Los Alamitos (2008)
- [8] Ralyté, J., Rolland, C.: An assembly process model for method engineering. In: Dittrich, K.R., Geppert, A., Norrie, M.C. (eds.) CAiSE 2001. LNCS, vol. 2068, pp. 267–283. Springer, Heidelberg (2001)
- [9] Harmsen, A.F.: Situational Method Engineering. Moret Ernst & Young, Amsterdam, The Netherlands (1997)
- [10] Ralyté, J., Rolland, C.: An Approach for Method Reengineering. In: Kunii, H.S., Jajodia, S., Sølvberg, A. (eds.) ER 2001. LNCS, vol. 2224, pp. 471–484. Springer, Heidelberg (2001)
- [11] Mirbel, I.: Connecting Method Engineering Knowledge: a Community Based Approach-IFIP WG8.1. In: Working Conference on Method Engineering, Geneva, Switzerland (2007)
- [12] Firesmith, D.G., Henderson-Sellers, B.: The OPEN Process Framework. An Introduction, Pearson Education Limited, Harlow, Herts, UK (2002)
- [13] Papazoglou, M.P., Traverso, P., Dustdar, S.: Service-Oriented Computing Research Roadmap (2006)
- [14] Papazoglou, M.P., Heuvel, W.J.: Service-Oriented Design and Development Methodology. International Journal of Web Engineering and Technology (IJWET) 2(4), 412–442 (2006)
- [15] Lane, S., Richardson, I.: Process Models for Service Based Applications: A Systematic Literature Review, Information and Software Technology, In Press (2010)
- [16] Arni-Bloch, N., Ralyté, J.: Service-Oriented Information Systems Engineering: A Situation-Driven Approach for Service Integration. In: Bellahsene, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 140–143. Springer, Heidelberg (2008)
- [17] Arni-Bloch, N., Ralyté, J.: MISS: A Meta-model of Information System Service. In: Proceedings of the 17th International Conference on Information System Development (ISD), Paphos, Cyprus, Springer, Heidelberg (2008)
- [18] Arni-Bloch, N., Ralyté, J., Léonard, M.: Service-Driven Information Systems Evolution: Handling Integrity Constraints Consistency. In: Persson, A., Stirna, J. (eds.) The practice of Enterprise Modeling. Proceedings of the 2nd IFIP WG 8.1 Working Conference, vol. 39(2009), pp. 191–206 (2009)
- [19] Henderson-Sellers, B., Ralyté, J.: Situational Method Engineering: State-of-the-Art Review. Journal of Universal Computer Science 16(3), 424–478 (2010)
- [20] Kumar, K., Welke, R.J.: Method engineering: a proposal for situation-specific methodology construction. In: Cotterman, Senn (eds.) Systems Analysis and Design: A Research Agenda, pp. 257–268. Wiley, Chichester, UK (1992)
- [21] Brinkkemper, S.: Method Engineering: Engineering of Information Systems Development Methods and Tools. Information and Software Technology 38(4), 275–280 (1996)

- [22] Saeki, M., Iguchi, K., Wen-yin, K., Shinohara, M.: A Meta-model for Representing Software Specification & Design Methods. In: Proceedings of IFIP WG8.1 Conference on Information Systems Development Process, pp. 149–166 (1993)
- [23] Rolland, C., Prakash, N.: A Proposal for Context-Specific Method Engineering. In: Brinkkemper, S., Lyytinen, K., Welke, R.J. (eds.) Method Engineering. Principles of Method Construction and Tool Support. Proceedings of IFIP TC8, WG8.1/8.2 Working Conference on Method Engineering, Atlanta, USA, pp. 191–208. Chapman and Hall, London (1996)
- [24] Brinkkemper, S., Saeki, M., Harmsen, F.: Assembly Techniques for Method Engineering. In: Pernici, B., Thanos, C. (eds.) CAiSE 1998. LNCS, vol. 1413, pp. 381–384. Springer, Heidelberg (1998)
- [25] Low, G., Mouratidis, H., Henderson-Sellers, B.: Using a Situational Method Engineering Approach to Identify Reusable Method Fragments from the Secure TROPOS Methodology. *Journal of Object Technology* 9(4), 91–125 (2010)
- [26] Henderson-Sellers, B., Serour, M., McBride, T., Gonzalez-Perez, C., Dagher, L.: Process construction and customization. *Journal Universal Computer Science* 10(4), 326–358 (2004)
- [27] Serour, M.K., Henderson-Sellers, B.: The role of organizational culture on the adoption and diffusion of software engineering process: an empirical study. In: Bunker, D., Wilson, D., Elliot, S. (eds.) *The Adoption and Diffusion of IT in an Environment of Critical Change*. IFIP/Pearson, pp. 76–88. Pearson, Frenchs Forest, Australia (2002)
- [28] Graham, I., Henderson-Sellers, B., Younessi, H.: *The Open Process Specification*. Addison-Wesley (1997)
- [29] ISO/IEC 24744. *Software Engineering – Meta-model for Software Development Methodologies*, ISO, Geneva (2007)
- [30] Firesmith, D.G., Henderson-Sellers, B.: *The Open Process Framework. An Introduction*. Addison-Wesley, London, UK (2002)
- [31] Object Management Group: *OMG Unified Modeling Language Specification, Version 1.4*, OMG (2002)
- [32] Henderson-Sellers, B., France, R., Georg, G., Reddy, R.: A method engineering approach to developing aspect-oriented modeling processes based on the Open process framework. *Information and Software Technology* 49(7) (2007)
- [33] Henderson-Sellers, B.: An open process for component-based development. In: Heineman, G.T., Councill, W. (eds.) *Component- Based Software Engineering: Putting the Pieces Together*, pp. 321–340. Addison- Wesley, Reading, MA, USA (2001)
- [34] Haire, B., Henderson-Sellers, B., Lowe, D.: Supporting web development in the OPEN process: Additional tasks. In: Haire, B., Henderson-Sellers, B., Lowe, D. (eds.) *Proceedings of 25th Annual International Computer Software and Applications Conference. COMPSAC 2001*, pp. 383–389. IEEE Computer Society Press, Los Alamitos (2001)
- [35] Henderson-Sellers, B., Haire, B., Lowe, D.: Using Open’s deontic matrices for e-business. In: Rolland, C., Brinkkemper, S., Saeki, M. (eds.) *Engineering Information Systems in the Internet Context*, pp. 9–30. Kluwer Academic Publishers, Boston, USA (2002)

- [36] Henderson-Sellers, B., Serour, M.: Creating a process for transitioning to object technology. In: Henderson-Sellers, B., Serour, M. (eds.) Proceedings Seventh Asia-Pacific Software Engineering Conference. APSEC 2004, pp. 436–440. IEEE Computer Society Press, Los Alamitos, CA, USA (2004)
- [37] Serour, M.K., Henderson-Sellers, B., Hughes, J., Winder, D., Chow, L.: Organizational transition to object technology: Theory and practice. In: Bellahsène, Z., Patel, D., Rolland, C. (eds.) OOIS 2002. LNCS, vol. 2425, pp. 229–241. Springer, Heidelberg (2002)
- [38] Henderson-Sellers, B., Hutchison, J.: Usage-Centered Design (UCD) and the open Process Framework (OPF). In: Constantine, L.L. (ed.) Performance by Design. Proceedings of USE2003, Second International Conference on Usage-Centered Design, pp. 171–196. Ampersand Press, Rowley, MA, USA (2003)
- [39] Petritsch, H.: Service-Oriented Architecture (SOA) vs. Component Based Architecture, http://petritsch.co.at/download/SOA_vs_component_based.pdf
- [40] Fahmideh Gholami, M., Habibi, J., Shams, F., Khoshnevis, S.: Criteria-Based Evaluation Framework for Service-Oriented Methodologies, UKSim. In: 12 th International Conference on Computer Modeling and Simulation, pp. 122–130 (2010)
- [41] Engels, G., Assmann, M.: Service-Oriented Enterprise Architectures: Evolution of Concepts and Methods. In: 12th International IEEE Enterprise Distributed Object Computing Conference. pp. xxxiv–xlxxx, IEEE Computer Society (2008)
- [42] Arsanjani, A., Allam, A.: Service-Oriented Modeling and Architecture for Realization of an SOA. In: IEEE International Conference on Services Computing, pp. 521–521 (2006)
- [43] Arsanjani, A., Ghosh, S., Allam, A., Abdollah, T., Ganapathy, S., Holley, K.: SOMA: a method for developing service-oriented solutions. IBM Systems Journal 47, 377–396 (2008)
- [44] Ralytė, J., Backlund, P., Kühn, H., Jeusfeld, M.A.: Method Chunks for Interoperability. In: 25 th International Conference on Conceptual Modeling. ER 2006, pp. 339–353 (2006)
- [45] Mirbel, I., de Rivières, V.: UML and the unified process. IRMA Press (2003)
- [46] Mirbel, I.: Connecting method engineering knowledge: a community based approach. Situational Method Engineering, pp.176-192 (2007)
- [47] Ralytė, J.: Towards situational methods for information systems development: Engineering reusable method chunks. In: Proceedings of ISD 2004, pp. 271–282 (2004)
- [48] Fahmideh, M., Jamshidi, P., Shams, F.: A Procedure for Extracting Software Development Process Patterns, In: Fourth UKSim European Symposium on Computer Modeling and Simulation, pp.75-83, Pisa, Italy, (2010)
- [49] SUN Microsystems, SOA RQ methodology - A pragmatic approach methodology.pdf, <http://www.sun.com/products/soa/soa>
- [50] Allen, P.: The Service-Oriented Process, in CBDi Journal <http://www.cbdiforum.com/reportsummary.php3?page=/secure/interact/2007-02/serviceorientedprocess.php&area=silver>
- [51] Erl, T.: Service-Oriented Architecture: Concepts, Technology, and Design Upper Saddle River: Prentice Hall PTR (2005)
- [52] Keith. M.: SOMA, RUP and RMC: the right combination for Service Oriented Architecture, IBM® Web Sphere® User Group, Bedford (2008)

- [53] Zimmermann, O.P., Krogdahl, C.: Elements of Service-Oriented Analysis and Design, IBM Corporation,
<http://www-128.ibm.com/developerworks/library/wssoad1>
- [54] Mittal, K.: Service-Oriented Unified Process (SOUP),
<http://www.kunalmittal.com/html/soup.shtml>
- [55] Jones, S.: A Methodology for Service Architectures, Capgemini UK plc -
open.org/committees/download.php/15071/AmethodologyforServiceArchitecturesASISContribution.pdf, <http://www.oasis>
- [56] Erradi, A.: SOAF: An architectural framework for service definition and realization. In: Proceedings of the IEEE International Conference on Services Computing, pp. 151–158. Chicago, USA (2006)

Conceptual Framework for Formalizing Multi-Agent Systems

Tawfig M. Abdelaziz

University of Garyounis, Faculty of Information Technology,
Department of Software Engineering
tawfigtawill@gmail.com

Abstract. Many of agent systems concepts are proposed in last decade. Most of them are inspired from the approach or discipline upon which they are based. Some of these concepts are developed based on the approach of extending existing Object Oriented concepts to include the relevant aspects of agents. Others are developed based on agent-based concepts or based on knowledge engineering concepts. The difference between disciplines of knowledge is causing the misunderstanding of some concepts that relate to agent systems, as well as some inconsistencies. In this paper, a conceptual formal framework is constructed to determine the essential MAS concepts and to reduce the possible turmoil and define the relationships among those concepts. The proposed framework is well-structured formal system that was constructed to ensure that the proposed MAS conceptual system is logically coherent and free of contradiction. Z language is used to represent this formal system.

Keywords: Multi-agent systems, conceptual framework, formal framework, Z-language.

1 Introduction

Recently, agent systems have proven to be a powerful new approach for designing and developing complex and distributed software systems. Developing such systems require a comprehensive and well-built agent-oriented methodology. Recently, many agent-oriented methodologies and modeling languages have been proposed such as MASD [1], Gaia [19,16], MaSE [9], MESSAGE [6], Tropos [5], HLIM [10], Prometheus [13], and AUML [3] etc. They are built and specifically tailored to the characteristics of agents. However, they are still considered incomplete and suffer several limitations for the following reasons: firstly; none of the existing agent-oriented methodologies has itself established as a standard nor have they been commonly accepted [12]. Furthermore, most of the concepts used by the agent-oriented methodologies, like agent roles, responsibilities, beliefs, goals, plans, and tasks do not have formal semantics or explicit formal properties. This is an important issue when these concepts are applied, as implementation constructs need to have exact semantics [8]. Moreover, Incomplete formality, despite, a number of approaches for formally specifying agent system concepts have been developed such as that by

Wooldridge [17,18] and Luck [12]. Until now, there was no complete formalism for MAS concepts, which were capable to clearly describe, specify and define them in an accurate manner. They were also unable to represent the important aspects of an agent-based system such as agent beliefs, goals, actions, and interactions. This is due to the lack of agreement amongst existing agent concepts [12]. Furthermore; most of the existing methodologies contain several misconceptions when introducing and defining the concepts and in building analysis and design concepts. This is due to the disagreement regarding agent concepts and terminology. There is in fact an extensive disagreement on the approaches that each methodology is based on [2]. Some methodologies work on the basis of AI approaches, others work on the basis of software engineering approaches, while others use both [15, 7].

In this paper, we tried to construct a conceptual framework for formalizing the MAS concepts and determine accurate tracks to build those concepts. The framework helps to reduce the possible confusion and the lack of contradiction apparent between them. It is aimed to ensure that the proposed MAS conceptual system is applicable, logically coherent and free of contradiction. This is done through introducing a well-structured formal system able to specify and represent MAS concepts, components, and behavior. This conceptual framework is exploited by developers as a foundation to build any agent oriented methodology concepts and models. This is done by transferring the concepts that are stated in this conceptual framework into models which represent the backbone of the new methodology. The overall objective of this paper is to describe MAS concepts and formalize them. In order that, we describe the MAS and propose the agent structure concepts that describe and define the agent system and related concepts such as roles, goals, beliefs, plans, tasks, interactions and dependencies.

In order to be able to do this formalization step, the following steps must be performed: Firstly; Identify why formalization is needed, define the requirements of the formal system, determine what the notation or formal language that will be utilized in the construction of this formalization, define MAS concepts and build a formal system for MAS concepts, and conclude all concepts and terminologies for the agent MAS and related concepts.

1.1 The Need for Formalization

In this section, the question why formalization is needed; is answered. There are many motivating reasons to utilize formalization in this dissertation. In general, the reasons are:

1. To link the proposed MAS concepts with the proposed methodology. In order to do that the following should be available:
 - A formal language that provides the link between MAS concepts and the proposed methodology. This is in order to make MAS concepts concrete and implementable. When a definition or a concept is developed formally; that means this definition will remain unchanged regardless of the situation in various contexts. There will be no change in the clarity in which it is understood. Hence, the best way to link between concepts and the methodology is the formality.

- A Formal language to enable the meaning of the concepts to be defined more precisely. The meaning of concept needs to be set on its correct place otherwise, the formality will be inaccurate.
- 2. Formalizing an agent system ensures that the concepts of the agent system are logically coherent and are free from contradictions.
- 3. Formality provides a precise and unambiguous language for specifying agent systems' concepts, components and behavior;
- 4. Formality addresses the needs of practical applications of agents. This is achieved by being capable of formally express most or even all aspects of an agency including but not limited to perception, action, belief, knowledge, goals, etc.
- 5. Formality helps identifying properties of agent systems against which implementations can be measured and assessed.
- 6. Formal system specification complements informal specification techniques.
- 7. Formal specifications are precise and unambiguous. They remove doubt in a specification.
- 8. Formal specification forces an analysis of the system requirements at an early stage. Correcting errors at this stage is cheaper than modifying a delivered system.
- 9. Formal specification techniques are most applicable in the development of critical systems and standards.

1.2 Formalization Requirements

We suppose that such a formal system should satisfy the following requirements:

- A formal system should precisely and unambiguously provide meanings for agent concepts and terms in order to be legible and understandable. The legible explicit notations enable moving from indistinct and conflicting informal understandings of models to a common conceptual system. A common conceptual system exists if there is a formal definition of the most important concepts involved in the class of models relevant to the MAS.
- The formal system should be well structured to provide a foundation for the development of new more refined concepts.
- The formal system should be capable to explicitly present, compare and evaluate the alternative designs of particular models and systems.

1.3 Formal Notations (Z Notation)

In this paper, Z-specification language (Z notation) is used as a notation to formalize the proposed multi-agent systems. The formal notation Z is based on set theory and predicate calculus, and has been developed at the Oxford University Computing Laboratory since the late 1970's [11, 4, 14, 20, 21]. Set theory includes standard set operators, set comprehensions, cartesian products, and power sets. The mathematical logic is a first-order predicate calculus. The propositional, predicate and modal logics are used. Together, they make up a mathematical (formal) language that is easy to learn and easy to apply. However, this language is only one aspect of Z. Another

aspect is the way in which the mathematics can be structured. Mathematical objects and their properties can be collected together in schemas or patterns of declaration and constraint. The schema language can be used to describe the state of a system, and the ways in which that state may change. It can also be used to describe system properties, and to reason about possible refinements of a design.

1.3.1 Z - Notation by Example .

In the following section a brief description of Z notation is presented with a small example. It explains the important symbols that are used in the formal system.

Z: Sets and Variables

- Declaring *SETS* in Z : $[Book], Response ::= ok / notok$
 These say that:
 - *Book* is a set; no implementation details needed/given
 - *Response* is a set containing only $\{ok, notok\}$
- Many predefined sets exist, e.g. \mathbb{N} (natural numbers)
- Can use many common operators on sets, e.g. \times, \mathbb{P} (cross product, powerset)
- Variables declared by *varname* : *setExpr*

e.g. "reference: $\mathbb{P} Book$ " means "reference is a member of the powerset of Book",
 i.e. "reference is a subset of Book"

The following table shows some important operators on sets.

Operator	Meaning
\mathbb{P}	Power set
\mathbb{P}_1	Non Empty Power set
F	Finite set
F_1	Non empty finite set

Z: Schemas

Schema is a structure describing some variables whose values are constrained in some way. It is like a C struct or Java interface, but also provides information about relationships between fields. A schema defines identifiers to their typed values. It consists of two parts: a declaration of variables; and a predicate constraining their values. We can write it in a form as follows.

Declarations
Predicates

Example:

$$Foo == [s,t : \mathbb{P}\mathbb{N} \mid s \subseteq t]$$

Meaning:

Foo is a schema containing two variables (s and t), which are both sets of natural numbers, such that $s \subseteq t$.

Multi-line form:

Foo
$s, t : \mathcal{P}\mathbb{N}$
$s \subseteq t$

[*Book, Borrower*] // Book and Borrower are two sets.

Library
<i>Borrowing</i> : $\mathcal{P}(\text{Borrower} \times \text{Book})$
<i>Reference</i> : $\mathcal{P}\text{Book}$
$\forall x : \text{Book} \bullet \forall y, z : \text{Borrower} \bullet ((y, z) \in \text{borrowing}) \wedge ((z, x) \in \text{borrowing}) \Rightarrow y = z$
$\forall x : \text{Book} \bullet \forall y : \text{Borrower} \bullet x \in \text{reference} \Rightarrow \neg (x, y) \in \text{borrowing}$

Meaning:

- Book and Borrower are sets.
- A library maintains info about which borrower is borrowing which books, which books are reference books.
- A book can be borrowed by a maximum of 1 unique borrower.
- No one can borrow reference books.

2 Formal System of MAS

One of the most suitable ways in which a mathematical notation can help achieving our goal is through the use of mathematical data types to model the data in a system. A notation of different types of logics is used to describe abstractly the effect of each operation of the system. This is in a way that makes it possible to reason about its behavior. The other main feature in Z is a way of decomposing a system specification into small pieces called schemas. By splitting the system specification into schemas, we can present it piece by piece. Each piece can be linked with a commentary, which explains informally the significance of the formal mathematics. We utilize this feature to represent MAS concept. In this section, we will define MAS concepts and formalize them. This step starts first with defining the concept with informal textual definition and then represents that concept formally (formal definition). This formal system starts with defining MAS concept and then defines its components such as agents, interactions, states etc. The following section describes a formal description for the MAS as an interactive environment and its components. Firstly, it describes the MAS environment and then describes its components in detail such as (Environment Attributes, States, Agents, Interactions, Dependencies, Events,

Triggers, and Actions). Furthermore, it describes the agent itself and its sub-components such as (agents attributes, agent Knowledge (goals, plans and beliefs) and agent capabilities (as a set of roles that the agent should perform in the system).

2.1 Multi-Agent System Concept

A **Multi-agent system** is a system composed of several agents, collectively capable of reaching goals that are difficult to achieve by an individual agents. A multi-agent system is a system showing the following characteristics (Jennings 1998):

- Each agent has incomplete capabilities to solve a problem.
- There is no global system control.
- Data is decentralized.
- Computation is asynchronous.

This system passes through a number of states as it operates. According to decomposition feature of Z language, the first step of our formal system is decomposing MAS into small schemas. It is then that the components of the MAS and their behavior can be presented.

2.2 MAS-Environment

Formally, we assume that the MAS is represented as an environment called MAS environment. The environment of MAS is a collection of sets that describe all the features of the system, its components and its behavior. Those sets affect each other. These components are stated as follows: the environment states, the agents that act on this MAS environment, the events that happen in the environment, the interactions that could take place between the agents, the dependencies between agents in the system and the attributes of the environment, which are the features of the MAS environment that are visible to the agents. For the purpose at hand, it will not matter what form these components (State, agent, interaction, dependency and event) take. These components are briefly described as a data type and they will be defined in detail afterwards. So we introduce MAS environment as a set of all environment states, set of agents and, set of interaction that could happen between agents, set of dependencies between the agents, set of events, set of actions, set of triggers, and set of environment actions as the following basic data types:

[State, Agent, Interaction, Dependency, Event, Triggers, Action]

State: The MAS environment state is a non empty set of attributes of the MAS environment. **Agent:** An agent is a system built to perform specific actions. This system is continuously changing its internal state and the state of the MAS environment. These states change when there are actions performed by the agents according to its knowledge. The agent knowledge represents the agent's beliefs about all entities that exist in the environment. It also represents the objectives it wishes to achieve. The change of the agent knowledge takes place through sensing the environment in which the agent resides. **Interaction:** is the way in which agents exchange information. Such exchange takes the form of message passing from one to

many agents. Interaction enables agents to cooperate and to coordinate in achieving their tasks. **Dependency**: is defined as the relationship of one task of the first agent (d) to another task of the second agent (successor) where the start of the task of the first agent (predecessor) is constrained by the start of the task of the second agent (successor). **Event**: can be defined as immediate occurrence that happens and changes at least one of the attributes of the MAS environment. **Environment action** is defined as an influence that happens in the environment and the agent is not responsible for that influence.

[EnvAction]

Trigger is a procedure that initiates an action (i.e. fires an action) when an event occurs in the environment or a change of the agent belief takes place. **Agent action** is the mechanism with which the agent performs its task in the system. Agent actions affect the environment, which in turn affects future decisions of agents. It is defined formally as a finite set of tasks that the agent performs in the system.

AgentAction == F₁ task

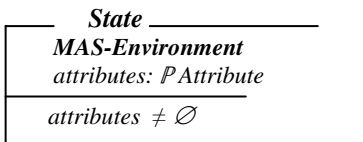
Action is a distinct task that can change the state of the MAS environment when performed. Actions are considered either Environment-Action or Agent-Action.

Action ::= EnvAction | AgentAction

This allows the sets to be named without saying what kind of objects they contain. The formal definition of each component or variable of the MAS that was exhibited in the previous declaration will be introduced in the following sub-sections.

2.2.1 MAS Environment State

MAS environment state is a complete and instantaneous description of the environment. It is like a very detailed snapshot of the environment. The states of MAS are dynamic, changes on its own accord and changes in response to interaction with the agent. Interaction means what the agents can perceive, what it can change, and what it can do. The state of environment is represented by the **Attribute** variable which must consist of a **non-empty** set of attributes.



Attribute: Environment attributes are simply features of the world and are the only characteristics that are visible by the agents in the system. An attribute is a perceivable feature.

2.2.2 Agent Concept

In this section, the agent definition is proposed in both informal and formal description. First we introduce the informal definition and then introduce the formal definition.

2.2.2.1 Agent definition. An agent is a persistent computer system that carries out some set of tasks on behalf of a user or other agents and is capable of:

- Function with some degree of autonomy (autonomy means the agent ability to work with minimum intervention by the real user. The autonomous agents have control over their tasks and resources and will take part in cooperative activities only if they chose to do so).
- Communicate with others (humans or agents) via specific agent communication language.
- Perceive its environment through sensors, act on the environment and react to the changes of the environment through effectors.
- Employ its knowledge to make decisions.
- Negotiate and coordinate with others (humans and agents) to achieve common goals.
- Gain knowledge from past experience to store the successful plans.
- Realize its goals by performing suitable roles.
- Be adaptable with the environment changes by responding in a timely fashion.
- Initiative (self starting).

2.2.2.2 Agent Formal Definition. An agent is a system defined as a set of attributes, a set of knowledge and a set of capabilities. Attributes are the only characteristics of the agent that appear. Knowledge is a set of beliefs and goals that the agent possess. Capabilities are a set of roles that the agent should perform in the system.

An introduction to the formal definition of the agent schema is presented in the following section.

[*Attributes, Knowledge, AgentCapabilities*]

Agent

attributes: \mathcal{P}_1 Attribute

knowledge: Knowledge

capabilities: AgentCapability

attributes $\neq \emptyset$

knowledge $\neq \emptyset$

capabilities $\neq \emptyset$

2.2.2.3 Attributes. Attributes are features of the agent, and are the only characteristics that are visible by other agents in the system.

2.2.2.4 *Agent Knowledge*. Agent knowledge can be defined as what each agent knows about the environment state; but also what each agent knows about other agent's knowledge. Agent knowledge represents the informational state of the agent about the environment including itself and other agents.

$Fact ::= proposition, FactState ::= true \mid false, Belief ::= (Fact, FactState)$

<p>Knowledge</p> <p>Agent</p> <p><i>beliefs</i> : $\mathcal{P} Belief$</p> <p><i>goals</i> : $\mathcal{P} Goal$</p> <p><i>changeofbelief</i>: $(Belief \times Event) \rightarrow Belief$</p>
<p><i>beliefs</i> $\neq \emptyset$</p> <p><i>goals</i> $\neq \emptyset$</p> <p>$\forall b: Belief \bullet \exists e: Event \bullet (changeofbelief(b,e)) \subseteq beliefs$</p>

It includes **Beliefs** and **Goals**. Formally, an agent's knowledge is represented by agent's beliefs and agent goals. For each belief there is an event that causes its value to change. **Beliefs** are facts that represent an agent's knowledge. They are used to accomplish an agent's behaviour. These facts are propositions that have been asserted to be either "true" or "false". The term "Fact" usually refers to a "ground proposition" i.e. a proposition that can be represented as a predicate applied to a sequence of instances or literals.

A **Belief** is a fact that is believed to be true about the working environment. An agent's belief is knowledge which constitutes a description of the world. An agent's belief may be taken to explicitly represent the agent's working environment or even about the agent itself or about other agents. Using the term **belief** - rather than **knowledge** - recognizes that what an agent believes may not necessarily be true and in fact may even change in the future.

$Task ::= seq AgentAction$

<p>Plan</p> <p>Goal</p> <p><i>tasks</i>: $\mathcal{P} Task$</p> <p><i>state</i> ::= $true \mid false$</p>
<p><i>tasks</i> $\neq \emptyset$</p> <p>$\exists p: Plan \bullet p \in plans$</p>

Goals are informational states of what it is planned to achieve. In order to formally describe the goal the following two questions need to be answered. When are goals initiated or started? And when are goals considered to be satisfied? The goal is started or initiated when its precondition(s) is satisfied. The goal is considered to be satisfied if and only if at least one of its plans is satisfied and its postcondition(s) is satisfied. (plans will discuss in the next section). **Plans** are an agent's view of the way a modeled agent will achieve its goals. As already mentioned, the goal is considered to be satisfied if and only if at least one of its plans is satisfied.

Condition ::= true | false

Goal

Knowledge

conditions : \mathbb{P} Condition

plans : \mathbb{P} Plan

achievegoal : Condition \rightarrow Goal $\rightarrow F_I$ Plan

plans $\neq \emptyset$

$\forall g : \text{Goal} \bullet \exists c : \text{Condition} \bullet (\text{achievegoal } c \ g) \subseteq \text{plans}$

$\forall \text{agt} : \text{Agent} \bullet \exists g : \text{Goal} \bullet (\text{agt} \models \Box g \Leftrightarrow \exists p \in g.\text{plans} \wedge p.\text{state} = \text{true})$

We read $A \models B$ as "A satisfies B", "A is satisfied in B", or "A forces B". The relation is called the satisfaction relation, evaluation, or **forcing** relation. The satisfaction relation is uniquely determined by its value on propositional variables. For each agent **agt** there exist a goal **g** such that for agent **agt** is considered to have satisfied goal **g** if and only if there is a plan **p** that belongs to goal **g** is satisfied.

2.2.2.5 *Agent Capabilities.* An agent capability means what the agent can perceive, what it can change, and what it can do in the multi agent system as a whole. Agent capabilities consist of a set of roles, a set of agent actions and a set of perceptions.

Perception == Action

AgentCapability

Agent

roles : \mathbb{P} AgentRole

agentactions : \mathbb{P} AgentAction

perceptions : \mathbb{P} Perception

agentact : Goal \rightarrow Perception \rightarrow AgentAction

canperceive : MAS-Environment \rightarrow Perception

roles $\neq \emptyset$

perceptions \subseteq capabilities

$\forall g : \text{Goal}; \exists p : \text{Perception} \bullet (\text{agentact } g \ p) \subseteq \text{capabilities}$

$\forall \text{masenv} : \text{MAS-Environment}; \text{actions} : \text{Action} \bullet \text{actions} \in \text{dom}(\text{canperceive } \text{masenv}) \rightarrow$

action = perceptions

Role is defined as a set of actions and activities that are assigned to or required or expected of an agent to be able to perform in the system. In other words, a role represents an agent behavior that is recognized, providing a means of identifying and placing an agent in a system. The distinction between an agent and a role is that an agent model describes characteristics that are inherent to an agent, whereas a role describes characteristics that an agent takes on. **Perceptions** represent incoming information from the environment to the agent. The agent reacts according to this information in term of actions. An agent perceives its environment through sensors that describe events or information on the state of the environment. These events or information trigger the agent to do actions that may update the agent's knowledge known as Beliefs and goals.

Trigger == Event | changeofbelief

<p>AgentRole</p> <hr/> <p>AgentCapability <i>responsibilities : P Action</i> <i>triggers : P Trigger</i> <i>triggeringactions: Trigger → Action</i></p> <hr/> <p><i>responsibilities ≠ ∅</i> $\forall agt : Agent \bullet (\exists role \in agt. roles \wedge role \neq \emptyset)$ $\forall role : AgentRole \bullet (\exists resp \in responsibilities \wedge resp \neq \emptyset)$ $\forall r \in responsibilities, \exists t \in triggers \bullet \diamond(triggeringactions(t) \subseteq capabilities)$</p>

Responsibilities of a role represent the main duties or tasks that the role performs to realize his objectives in the system. For each agent there is at least one role that should be performed in the system. And for each role there is at least one responsibility that should be performed by this role. For each responsibility, there exists a trigger (to be explained in detail in section 2.2.6), which is possibly triggering an action that belongs to the agent capabilities.

2.2.3 Events

Events are perceived and afterwards processed by agents in order to launch which plans or goals should be selected to achieve. An agent may react to events that change its knowledge. Events change the agent's knowledge because its perception of the environment has changed. A triggering event defines which events may lead to the execution of a particular plan to achieve a particular goal.

<p style="text-align: center;"><i>[EAction, Location, Time]</i></p> <p>Event</p> <hr/> <p>MAS-Environment <i>action : EAction</i> <i>location : Location</i> <i>time : Time</i> <i>triggerevent: Action → Event</i></p> <hr/> <p><i>action ≠ ∅ ∧ location ≠ ∅ ∧ time ≠ ∅</i> $\forall e : Event \bullet \exists t, t_1 : Time \bullet HoldsAt(e, t) \Leftrightarrow [\exists a : Action \bullet Happens(a, t_1) \wedge Initiates(a, e, t_1) \wedge (t_1 < t)]$ $\forall e \in Event \bullet \diamond \exists a \in Action (triggerevent a) \subseteq events$</p>

An Event is an action that happens at a given place and time. According to this definition we can consider the event as a structure composed of action, location and time. Action uniquely identifies the event. Location is that place where the event happens. Time is that time when the event happens.

HoldsAt (e, t) means the event e holds at time t . Happens (a, t_1) means action a happens at time t_1 . Initiates (a, e, t_1) means action a initiates event e at time t_1 . The predicate part means the event e is true at time t if : An action a has taken place by $Happens(a, t_1)$, this took place in the past $t_1 < t$, and the action has an effect to the event e by $Initiates(a, e, t_1)$.

2.2.4 Agent Interactions

Interactions are the way how agents exchange information. Such exchange amounts to message passing from one agent to many agents. Interaction enables agents to cooperate and to coordinate in achieving their tasks. Protocols : **A Protocol** is a sequence of rules which guide the interaction that take place between several agents. These rules determine the format and transmission of messages exchanged between agents. These rules define what messages are possible for any particular interaction state. The set of possible messages is finite. Messages: **A Message** is a unit of information or data that is transmitted from one agent to another. A message can be defined formally as any information sent as an agent interacts with another.

Protocol == Seq rules

<p style="text-align: center;">Interaction</p> <hr/> <p>MAS-Environment</p> <p><i>origagent == Agent {first_agent}</i></p> <p><i>destagent == Agent {second_agent}</i></p> <p><i>interactingagents: Agent × Agent</i></p> <p><i>protocols: P Protocol</i></p> <p><i>messages : P Message</i></p> <p><i>interpretmessage : Message → Protocol</i></p> <hr/> <p><i>messages ≠ ∅</i></p> <p><i>protocols ≠ ∅</i></p> <p>$\forall m : Message \bullet (origagent \neq destagent)$</p> <p>$\forall i \in interactions \bullet \square (i.origagent \in interactingagents \wedge$ $i.destagent \in interactingagents)$</p> <p>$\forall i : Interaction \exists m : Message \bullet (interpretmessage m) \subseteq protocols$</p>

2.2.5 Agent Dependency

Dependency is a relationship between two agents, dependee and dependant. The dependant agent depends on another agent (the dependee) to do or provide something (dependum). There are three types of dependencies (dependum) between a dependant and a dependee namely: Resource, Goal and Task dependencies.

$Task == AgentAction, Dependum == Goal \mid Task \mid Resource,$
 $Dependant == Agent, Dependee == Agent$

Dependency

MAS-Environment

$dependum : Dependum$
 $depender : \mathbb{F}_1 Dependant$
 $dependee : \mathbb{F}_1 Dependee$

$\forall agt1 : Dependant \bullet \exists agt2 : Dependee \bullet (agt1 \neq agt2)$

$\forall agt2 : Dependee \bullet (\exists d : Dependum \wedge d = Resource) \Rightarrow (agt2 \in Dependee \wedge agt1 \in Dependant)$

$\forall agt1 : Dependant ; agt2 : Dependee \bullet \exists d : Dependum \bullet$

$(d = Task \wedge Depends(agt1, agt2, d)) \Leftrightarrow$

$(d \notin agt1.capabilities.agentactions \wedge d \in agt2.capabilities.agentactions)$

$\forall agt1 : Dependant ; agt2 : Dependee \bullet \exists d : Dependum \bullet$

$(d = Goal \wedge Depends(agt1, agt2, d)) \Leftrightarrow (d \notin agt1.knowledge.goals \wedge d \in agt2.knowledge.goals)$

2.2.6 Triggers

A trigger is an environmental stimulus which stimulates an agent to detect events or to detect changes of an agent's beliefs in the system and react by initiating some specific action. An agent perceives its environment through triggering information that describe events or describes changes of the agent beliefs on the state of the environment. These triggering information could result e.g. from actions that are observed by the agent, or receipt of a message from another agent.

MAS-Environment

$attributes : \mathcal{P} Attribute$
 $states : \mathcal{P} State$
 $agents : \mathcal{P} Agent$
 $events : \mathcal{P} Event$
 $actions : \mathcal{P} Action$
 $interactions : \mathcal{P} Interaction$
 $dependencies : \mathcal{P} Dependency$
 $changestate : (State \times Event) \rightarrow State$
 $trigger : Event \rightarrow Action$

$attributes \neq \emptyset$

$agents \neq \emptyset$

$states \neq \emptyset$

$\forall s : State \bullet \exists e : Event \bullet (changestate(s, e) \subseteq states)$

$\forall action \in actions \bullet \exists event \in events \bullet \square(action \Rightarrow event)$

$\exists event : Event \bullet \exists action : Action \bullet (trigger(event)) \subseteq actions$

$\forall d : Dependency \bullet (d.depender \in agents \wedge d.dependee \in agents)$

The perception of new triggering information stimulates the agent to perform tasks and actions that update the agent's knowledge or beliefs. This may be thought of as a collection of triggering information constituting the agent's triggers model of the world. Agents perceive triggers/information entities. These triggers carry two main classes of information, **Events**: these are objects describing events that inform the agent that something has happened; and **Change of Beliefs**: this refers to the state of entities in the Agent's world. Objects of this type inform the agent about some information; for example the state of some resource parameter.

The trigger concept has implicitly defined formally within some concepts defined above such as roles and events. Therefore, there is no need for a formal definition, but we will be content with informal definition.

3 Conclusion

We propose an interpreted and comprehensive model called the MAS conceptual framework. This framework introduces specific criteria to define agent concepts. This is done by creating a rational model that exhibits the relationships between concepts, and the criteria for selecting concepts. This will enable us to specify the requirements and essential properties of the software environment. The main goal of this framework is to define MAS as a conceptual system with matching property to exploit any system targeted for analysis according to the conceptual system properties. In this paper, we generalized a definition capturing the concept of an agent in a general-context and its derived and related concepts in the same context. The definition is to show an agent (software) as a logical cognitive computational (LCC) model. This model describes behavioral and structural qualities of software that acts in a relationship of agency. Furthermore, we proposed the MAS organizational structure to select MAS coordination and cooperation under constraints of the target obtained architecture. Moreover, we have shown how the classification described by the proposed conceptual framework satisfies each of the requirements stated in section 1.0. First, it provides clear and precise definition for agent and MAS that allow a better understanding of the functionality of different systems. Second, it provides a foundation of subsequent development of more refined concepts. Finally, it should also enable alternative designs to explicitly presented and compared. In addition, we have shown MAS to be a complete conceptual system with some degree of adaptation. Finally, we have shown how Z notation has enabled us to produce specification that generally accessible to researchers in AI, software engineering as well as practitioners of formal systems. Through the use of schema inclusion, we are able to describe our conceptual framework at the highest level of abstraction and then by incrementally increasing the detail in specification. Z does not restrict us to specific mathematical logic, but instead provides a general mathematical framework with in different models. In particular, Z allows us to extend the framework and refine it further to include a more varied and more inclusive set of concepts.

References

1. Abdelaziz, T.M., Elammari, M., Unland, R., Branki, C.: MASD: Multi-Agent Systems Development Methodology, *Multi-agent and Grid Systems Journal* (February 2010) ISSN 1574-1702
2. Abdelaziz, T.M., Unland, R., Elammari, M.: A Framework for the Evaluation of Agent-oriented Methodologies, In: 4th Int. Conf. on Innovations in IT, Dubai, UAE (2008)
3. Bauer, B., Odell, J.: UML 2.0 and agents: how to build agent-based systems with the new UML standard. *Journal of Engineering Applications of AI* 18 (2) (2005)
4. Bowen, J.: Formal Specification and Documentation using Z: A Case Study Approach (2003), For further on-line information see <http://www.afm.sbu.ac.uk/zbook/>
5. Bresciani, P., Giorgini, P., Giunchiglia, F., Mylopoulos, J., Perini, A.: Tropos: An agent-oriented software development methodology, Technical Report DIT-02-0015, Uni. of Trento, Dep. of Information and Communication Technology (2002)
6. Caire, G., Leal, F., Chainho, P., Evans, R., Jorge, F.G., Pavon, J.G., Kearney, P., Stark, J., Massonet, P.: Methodology for agent-oriented software engineering, Technical Information Final version, European Institute for Research and Strategic Studies in Telecommunications (EURESCOM), Project, p. 907 (2001)
7. Dam, K.H., Winikoff, M.: Comparing Agent-Oriented Methodologies. In: The International Bi-Conference Workshop on Agent-Oriented Information Systems, AOIS (2003)
8. Dastani, M.: AgentLink-III Technical Forum Group, PROMAS, Report (2004)
9. DeLoach, S.A.: The MaSE Methodology. In: The Agent-Oriented Software Engineering Handbook Series: Multi-agent Systems, Artificial Societies, and Simulated Organizations, vol. 11, Springer, Heidelberg (2004)
10. Elammari, M., Lalonde, W.: An Agent-Oriented Methodology: High-Level and Intermediate Models HLIM. In: Proceedings of AOIS, Heidelberg (1999)
11. Hayes, I.J. (ed.): Specification Case Studies, 2nd edn. Prentice Hall International Series in Computer Science (1993)
12. Luck, M., McBurney, P., Preist, C.: Agent Technology: Enabling Next Generation Computing (A Roadmap for Agent Based Computing), AgentLink (2005)
13. Padgham, L., Winikoff, M.: Prometheus: A methodology for developing intelligent agents. In: Third International Workshop on Agent-Oriented Software Engineering (2002)
14. Spivey, J.M.: The Z Notation: A Reference Manual, 2nd edn. Prentice Hall International Series in Computer Science (1992)
15. Sturm, A., Shehory, O.: A Framework for Evaluating Agent-Oriented Methodologies. In: Workshop on Agent-Oriented Information System (AOIS), Melbourne, Australia, AOIS (2003)
16. Wooldridge, M.J., Jennings, N.R., Kinny, D.: The Gaia methodology for agent-oriented analysis and design. *Autonomous Agents and Multi-Agent Systems* 3(3) (2000)
17. Wooldridge, M.: The Logical Modelling of Computational Multi-Agent Systems, PhD thesis, Department of Computation, UMIST, Manchester, UK (1992)
18. Wooldridge, M.: Introduction to Multi-agent Systems. John Wiley and Sons, Chichester (2002)
19. Zambonelli, F., Jennings, N.R., Wooldridge, M.: Developing multi-agent systems: The Gaia methodology. *ACM Transactions on Software Engineering and Methodology* 12(3), 317–370 (2003)
20. Z archive, Oxford University Computing Laboratory (1995), <http://www.zuser.org/z/>
21. Z FORUM, Oxford University Computing Laboratory, 1986 onwards. Electronic mailing list: vol. 1, 1–9 (1986), vol. 2, 1–4 (1987), vol. 3, 1–7 (1988), vol. 4, 1–4 (1989), vol. 5, 1–3 (1990)

Mining Optimal Utility Incorporated Sequential Pattern from RFID Data Warehouse Using Genetic Algorithm

Barjesh Kochar¹ and Rajender Singh Chhillar²

¹H.O.D (IT), GNIM & Research Scholar,
Department of Computer Science & Applications, M.D.U Rohtak, India
barjeshkochar@gmail.com

²Professor & Former Head of Department of
Computer Science & Applications, M.D.U Rohtak, India
chhillar01@rediffmail.com

Abstract. Today, identification of sequential patterns from a huge database sequence is a major problem in the field of KDD. In addition, if the entire set of sequential patterns existing in a large database is presented, the user may find it difficult to understand and employ the mined result. In order to overcome these issues, we propose an efficient data mining system to generate the most favorable sequential patterns. The proposed technique first generates datasets from the warehoused RFID data. Each mined pattern has distinct utility and the most favorable sequential patterns are generated from the mined sequential patterns by using Genetic Algorithm (GA). A fitness function is used in GA to find out the sequential pattern that provides maximum profit. The implementation result shows that the proposed mining system accurately extracts the important RFID tags and its combinations, nature of movement of the tags and the optimum sequential patterns.

Keywords: Data Mining System, RFID, Genetic Algorithm (GA), Sequential Pattern Mining, Fuzzy rules.

1 Introduction

Huge amount of data collection has been possible throughout the previous decades, because of the recent development of information, and accessibility of low-priced storage. Analyzing this data and utilizing this information for comprehending competitive benefits, is the eventual purpose behind this huge data collection i.e., determining previously unknown patterns in data that can assist in the decision making process [1] [8]. Generally, conventional data analysis methods cannot be employed for large data sets because they are based on direct handling of data by humans. Database technology has got the basic tools for competent storage and searching of large data sets. But, the challenging and unsettled issue assists humans in analyzing and understanding large masses of data. The emerging data mining field has the potential to meet these challenges as they provide novel techniques and intelligent tools. The term data mining, which is also called Knowledge Discovery in Databases

(KDD), is defined as “the non-trivial extraction of implicit, previously unknown, and potentially useful information from data” [2] [7].

Data mining [3] effectively used for converting huge volumes of data into small valuable pieces is a multidisciplinary united effort that involves machine learning, and statistics. In a real-world application, the ultimate goal of a data mining task is to allow a company to either improve its marketing, sales, and customer support operations or to identify a deceitful customer through better understanding of its customers. Data mining methods have been successfully carried out in a variety of fields including marketing [10], manufacturing, process control, and fraud detection [9], bioinformatics, information retrieval, adaptive hypermedia, electronic commerce and network management [4]. Data mining tasks are of two types, namely descriptive and predictive [5]. Descriptive mining portrays the fundamental characteristics or common properties of the data in the database. The technique of predictive mining interprets patterns present in the data which assist in making predictions. Tasks like Classification, Regression and Deviation detection are involved in predictive mining methods.

Many latest and emerging applications require information to be mined from a huge database. One such field is Radio Frequency Identification (RFID), which incorporates sequential pattern mining in the RFID database. Radio Frequency Identification (RFID) is a high-speed, real-time, precise information gathering and processing technology, which distinctively recognizes the objects by utilizing radio-frequency signal [6]. RFID technology can be employed for achieving considerable productivity gains and efficiencies in an extensive variety of organizations and individuals including hospitals and patients, retailers and customers, and manufacturers and distributors all through the supply chain [11]. Long sequences present in text data, biological data, software engineering, and sensor networks have motivated the study of mining repetitive gapped subsequences to capture the occurrences of sequential patterns repeating within each sequence of a large database and then use them as features for classification or prediction. The tags are very diverse from printed barcodes in their ability to hold data, and capability of reading tags from much greater distances and also when they are not in line-of-sight [12].

Finding all frequent sequential patterns with a user-specified least support is the goal of sequential pattern mining. Usually, sequential pattern mining approaches are either generate-and-test (also known as Apriori) approaches or pattern growth (also known as divide-and-conquer) approaches or vertical format method approaches [13]. Most of the numerous approaches [15] that are proposed for sequential pattern mining are focused on the following two issues: (1) enhancing the competency of the mining process and (2) widening the mining of sequential patterns to other types of time related patterns [16]. The retailing industry problem motivates the issue of sequential patterns discovery. However, the results are applicable to numerous scientific and business domains, like stocks and markets basket analysis, natural disasters (e.g. earthquakes), DNA sequence analyses, gene structure analyses, web log click stream analyses, and so on [18]. Time is the most important feature for this task, mainly when the results are required in a limited period of time [17].

In many cases, sequential pattern mining still faces hard challenges in both efficacy and competence, though efficiency of mining the whole set of sequential patterns has been enhanced considerably. Though a huge quantity of sequential patterns exist in a

large database, only a small subset of such patterns often interests a user. Presenting the complete set of sequential patterns would make the mining result tough to understand and hard to employ [22]. For optimizing the cost of the interesting sequential patterns Genetic Algorithm (GA) is employed. GA employs processes such as genetic combination, mutation, and natural selection in a structure, based on the concepts of evolution. GA optimizers are vigorous and they function well with discontinuous and non differentiable functions, where the customary local optimizers fail.

Though efficient algorithms that have been proposed for mining, mining large amount of sequential patterns from huge databases continues to be a computationally expensive task. In this work, we propose an effective data mining system for generating the optimum sequential pattern. The main aim of this research is to develop a RFID data mining technique to discover an optimum sequential pattern based on utility. The rest of the paper is organized as follows: section 2 describes some of the recent related works. Section 3 briefs about GA and section 4 details about the proposed method and optimization of sequential patterns using GA. Experimental results and analysis of the proposed methodology are discussed in Section 5. Finally, concluding remarks are provided in Section 6.

2 Related Works

Numerous researches have been proposed by researchers for the purpose of effective data mining. In this section, a brief review of some such important contributions from the existing literature is presented.

J. Hu and A. Mojsilovic [18] have proposed an algorithm for frequent itemset mining to identify high-utility item combinations. Unlike the customary association rule and frequent item mining methods, the objective of their algorithm has been to locate segments of data, defined through combinations of some items (rules), which gratify certain conditions as a group and maximize a predefined objective function. They have devised the task as an optimization problem, and presented a competent estimation to resolve it by specialized partition trees, called High-Yield Partition Trees, and examined the functioning of diverse splitting strategies. The algorithm when tested on “real-world” data sets has yielded very good results.

Jian Pei et al [19] have described that constraints are vital for numerous sequential pattern mining applications. But systematic study on constraint-based sequential pattern mining has not been available. In their paper, the issue has been investigated and they have shown that the framework developed for constrained frequent-pattern mining is not a perfect suit for their mission. On the basis of a sequential pattern growth methodology an extended framework has been developed. Their study has illustrated that under the new framework the constraints could be effectively and efficiently pushed deep into sequential pattern mining. Furthermore, their framework could also be extended to constraint-based structured pattern mining.

Shigeaki Sakurai et al. [20] have described that all frequent sequential patterns included in sequential data has been efficiently found out by sequential mining methods. Those methods have assessed the frequency by utilizing the support, which is the previous criterion that satisfies the Apriori property. However, the discovered

patterns have not been always consistent with the interests of the analysts because the patterns have been general and the analysts have not been capable of acquiring new knowledge from them. They have proposed a criterion called sequential interestingness to find out sequential patterns that are more attractive for the analysts. They have demonstrated that the Apriori property has been satisfied by the criterion and the relationship between the criterion and the support. Furthermore, based on the criterion, they have proposed a competent sequential mining technique. Finally, they have employed their proposed technique on two types of sequential data and proved its effectiveness.

Themis P. Exarchos et al. [21] have proposed a two processes methodology for sequence classification by utilizing sequential pattern mining and optimization. In the first stage, a sequential pattern has been defined based on a set of sequential patterns, and two sets of weights one has been introduced for the patterns and the other has been introduced for the classes. In the second stage, the weight values have been assessed by employing an optimization technique for achievement of best classification precision. By altering the number of sequences, the number of patterns and the number of classes, extensive appraisal has been carried out on the methodology, and it has been compared with similar sequence classification approaches.

S. Shankar et al. [22] have described that the data mining process produces numerous patterns from a given data has been a well accepted fact. The process of discovering frequent item sets and association rules has been the most important task in data mining. For mining frequent item sets and association rules several competent algorithms have been available in the literature. In recent years incorporating utility considerations in data mining tasks has been gaining popularity. The business value has been improved by certain association rules and these rules of interest have been acknowledged for a long time by the data mining community. The discovery of frequent item sets and association rules from transaction databases has been beneficial to numerous business applications. A complete survey and study of a variety of existing techniques for frequent item set mining and association rule mining with utility considerations has been presented in their paper.

Mourad Ykhlef and Hebah ElGibreen [23] have described that mining sequential patterns in large databases has become a vital data mining task with broad applications. Mining sequential patterns that describe the potential sequenced relationships among items in a database has been a significant task in the data mining field and numerous distinct algorithms have been introduced for that task. Though the correct optimal Sequential Pattern rule could be found by traditional algorithms the time taken by them has been considerable especially when they are applied on large databases. In those days, certain proposed evolutionary algorithms, like Particle Swarm Optimization and Genetic Algorithm have employed to tackle that problem. They have proposed a new variety of hybrid evolutionary algorithm that enhances the pace of convergence of the evolutionary algorithms by combining Genetic Algorithm (GA) with Particle Swarm Optimization (PSO) for mining Sequential Pattern. Their algorithm has been referred to as SP-GAPSO.

Jyothi Pillai and O.P.Vyas [24] have described utility mining as the emerging topic in the field of data mining. Recognizing the itemsets with highest utilities by taking into consideration the profit, quantity, cost or other user preferences has been the

chief purpose of Utility Mining. Finding itemsets that have utility above a user-specified threshold has been the purpose of mining High Utility itemsets from a transaction database. Itemsets that takes place regularly has been determined by Itemset Utility Mining which is an extension of Frequent Itemset mining. High-utility itemsets contain rare items in several real-life applications. Valuable information has been provided by rare itemaets in diverse decision-making domains including business transactions, medical, security, fraudulent transactions, and retail communities. For example, in a supermarket, microwave ovens or frying pans have been rarely purchased by customers compared to bread, washing powder and soap. But the profit contributed to the supermarket by the former has been more compared to the latter. Likewise, in several application fields, high-profit rare itemsets have been recognized to be highly beneficial. Lots of researches have been proposed for the purpose of itemset utility mining and they have presented an analysis of diverse such high utility rare itemset mining algorithms existing in the literature.

3 Genetic Algorithm (GA)

A search and optimization technique inspired by nature's evolutionary processes is genetic algorithm (GA). A population of candidates is iterated through multiple generations of selection, crossover, and mutation until an optimized solution is obtained, much in a manner similar to that of the "survival of the fittest". GA is a computer based optimization technique that employs the Darwinian evolution of nature as a model [24]. The work of Holland (1975) obtained a huge popularity for them. Usually, they are employed for problems, which have an immense and complex search space with an increased number of local optimums [27]. The strength behind GAs is the fact that the search space is traversed in parallel by arbitrarily generating solutions and those solutions are endlessly evaluated with a fitness function [25]. Generally, three different search phases are there in GA: (1) creating an initial population, (2) Evaluating the population by a fitness function and (3) producing a new population [21]. In GA, the solutions are termed as individuals or chromosomes [27]. The genetic search starts with an arbitrarily generated population inside which, a fitness function evaluates every individual.

The individuals of existing and following generations are duplicated or eliminated on the basis of the fitness values. By applying GA operators further generations are produced [21] i.e. reproduction, crossover and mutation are sequentially applied to each individual with certain probabilities [23], [22]. The first operator which is the production operator (elitism) produces one or more copies of any individual that posses a high fitness value; or else, the individual is detached from the solution pool [29]. Two randomly chosen parent individuals are taken by the crossover operator as input, and they are combined to generate two children individuals. This process of combining takes place by choosing two crossover points in the strings of the parents and then exchanging the genes between these two points [26]. The mutation of individuals through the alteration of parts of their genes is the next step in each generation [30]. Mutation brings inconsistency into the population of the succeeding generation by altering a gene of a chromosome. Making sure that the search algorithm is not bound on a local optimum is its main goal [22]. It is used to make sure that all

likely alleles can go into the population and hence preserve the population diversity [21]. It is a very important component for GAs and it is a variation operator which produces diversity for GAs [28].

4 An Efficient Data Mining System Based on GA

By means of a novel data cleaning, transformation and loading technique the RFID data is effectively warehoused, in a database dedicatedly created for storing the RFID data. The previous work illustrated that the required knowledge from the warehoused RFID data was efficiently mined by the proposed novel RFID data mining system. In a large database, large numbers of sequential patterns are mined, but a user is often interested in only a small subset of such patterns. Therefore presenting a complete set of sequential patterns may make the mining result hard to understand or use. Although efficient algorithms have been proposed, mining a large amount of sequential patterns from large data sequence database is inherently a computationally expensive task. So, the present work is intended to discover an optimum utility assigned sequential pattern in terms of cost. To identify the optimal sequential pattern a GA-based technique is employed. After the fuzzy rules are created from the sequential patterns, the optimal sequential patterns are recognized by the GA based method as per their assigned utility. The sequential pattern with maximum profit is discovered using the fitness function of the GA. For easy understanding of the proposed mining system before detailing the proposed mining system, the optimal sequential pattern of RFID data is briefed in the following sub-section.

Let R_i ; $0 \leq i \leq N_R - 1$ be the number of RFID readers that are available and T_j ; $0 \leq j \leq N_T - 1$ be the RFID tags which are in movement, where, N_R is the number of readers and N_T is the number of Tags that are present in the warehouse. During the process the tags may enter into any reader at any time. Each RFID tag has its own Electronic Product Code (EPC), which represents the class to which T_j belongs. The Tag representation with EPC code can be expressed as T_{kj} ; $k \in [EPC_1, EPC_{N_{cl}}]$, where, N_{cl} is the number of product classes available.

In the proposed data mining system, the first process is processing the warehoused data and generating an I-dataset. By querying the warehoused dataset, I-dataset is generated. So as to extract the required path information from the I-dataset, the I-dataset is subjected to sequential pattern mining. A generalized Tag ID is employed in the I-dataset and set $\{P\}$. But, the Tag IDs are practically represented with EPC code. So, from this moment onwards, the Tag IDs are grouped as per the product with which the tags are attached. From the pattern set $\{S\}$ the sequential patterns are attained. Each pattern has its own support, i.e. frequency of incidence, which is attained in the support set sup . From the attained sequential patterns, S_l with support greater than the minimum support are selected for additional mining

operations and the remaining patterns which do not convince this condition are removed. Then, the sequential patterns which have the greater support are mined.

Each pattern has its own support, i.e. frequency of occurrence of the pattern, which is obtained in the support set sup . From the obtained S_l (as $L = 1$), the sequential patterns with support greater than sup_{\min} ($\text{sup} \geq \text{sup}_{\min}$) are selected for further mining operations and the remaining patterns that doesn't satisfy sup_{\min} are eliminated. Then, the sequential patterns with $L > 1$ are mined. The pseudo code for mining the sequential patterns with $L > 1$ is given in Fig. 2. The mined sequential patterns using the algorithm, which is illustrated in the pseudo code, are checked for the minimum support. The sequential patterns that have support greater than the minimum support sup_{\min} are selected for the further process.

```

for L = 2 to Np
    for every L - length combination
        Sx1x2...xL(L) ← Sx1(1) ∩ Sx2(1) ∩ ... SxL(1) : {xi; 1 ≤ i ≤ L} ∈ (0, Nc + 1)
        for y = 0 to |Sx1x2...xL(L)|-1
            supx1x2...xL(L)(y) ← min(supx1(1)(z), supx2(1)(z) ... supxL(1)(z)):
                Sxi(1)(z) - Sx1x2...xL(L); z ∈ [0, |Sxi(1)|-1]
        end for
    end for
end for

```

Fig. 1. Pseudo code for mining sequential patterns with $L > 1$ from the I-dataset

From the sequential patterns that are mined by the mining algorithm with different combinations, fuzzy rules are produced. Therefore, from the mining algorithm different kinds (length) of pattern sets are attained and several sequential patterns

$\left| S_{x_g}^{(L)} \right|$ are present in each pattern set. A set of support values are also obtained for each kind of pattern set, which comprises the support values for every sequential pattern that is present in the pattern set. Then on the basis of sequential pattern sets, the fuzzy rules are generated. Generally, the fuzzy rules are of the form, *if* $IN_1 = i_1, IN_2 = i_2, \dots, IN_n = i_n$ *then* $OUT = i_{out}$. In the fuzzy rules generation process, a similar format is employed. The nature of the tag movement with a fuzzy score is described by the fuzzy rules obtained from the system. A part of the tag is given the remaining path of the tag (product) is held by the fuzzy rules.

Once the tags are cleaned and mined using effective techniques, then the data warehouse contains large number of sequential patterns. But the users are not interested in using all the patterns which are available in the database. So selecting the user interesting sequential patterns is a necessary one in the data warehouse management. Here we proposed the system which effectively identifies the user interested patterns from the mined database. After the fuzzy rules are generated, the optimal sequential pattern is identified by using GA.

4.1 Optimization of Sequential Patterns Based on Their Utility

Let k be the number of fuzzy rules generated from the preceding work. Along with the fuzzy rules, fuzzy scores are generated from the extracted sequential patterns. The main aim of the proposed system is to optimize the sequential pattern, i.e., extract the most profitable sequential patterns from the list available. By employing GA, we must optimize the sequential patterns based on its utility and frequency. In general, each product has its own utility and the utility of the i^{th} product is represented as Ut_i . From the set of fuzzy rules that are generated with the combination of products and frequency, the optimum sequential pattern can be recognized by employing the GA. GA is an alternate of the stochastic beam search in which the successor states are generated by the combination of two parent states instead of modifying a single state.

The analogy of natural selection is similar to stochastic beam search. Let R_k^S be the set of fuzzy rules generated and fqi be the frequency of each rules in the set R_k^S .

Initialization: GA starts with an initial population composed of a set of randomly generated states or individuals. Each Individual is nothing but a possible solution of the problem. Each individual’s chromosome is made up of a combination of its parameters or “genes”. The “solution space” contains all the individuals which belong to the problem. The initial population of solutions is formed by a defined number of randomly-generated individuals. The selection process begins with the development of a fitness function for calculating the fitness of an individual to be the solution of the problem. A chromosome is formed by the collection of P number of genes.

Primarily a population contains q number of chromosomes. First N_p numbers of chromosomes each of length N_{Tx} are randomly selected. The randomly generated chromosomes can be represented as follows,

$$X^{(j)} = [x_0^{(j)} \ x_1^{(j)} \ x_2^{(j)} \ \dots \ x_{N_{Tx}-1}^{(j)}] \ ; \ 0 \leq j \leq N_p - 1 \tag{1}$$

Each gene of the chromosomes is assigned either a ‘1’ or a ‘0’ with the aid of the decision making process as follows

$$x_n^{(j)} = \begin{cases} 1 & ; \text{if } n^{(j)} \in [P^{(j)}] \\ 0 & ; \text{else} \end{cases} \tag{2}$$

In Eqn. (2), $x_n^{(j)}$ is the n^{th} gene of the j^{th} chromosome, where, $0 \leq n \leq N_{Tx} - 1$. The vector $[P^{(j)}]$ which represents the position of each gene is obtained as follows

$$[P^{(j)}] \ll r_a^{(j)} \% N_{Tx} \quad (3)$$

In Eqn. (3), $r_a^{(j)}$ indicates the a^{th} random integer for the j^{th} chromosome, where $0 \leq a \leq n_{Tx} - 1$.

Then fitness of each of the N_p chromosomes is evaluated using the fitness function

Fitness: Fitness function is a unique kind of objective function that prescribes the optimality of a solution (that is, a chromosome) in a GA. It is used for assigning rank for the chromosomes. An improved generation (hopefully) will be generated by allowing breeding and mixing of the datasets of those chromosomes that are either optimal or at least more optimal compared to other chromosomes. In our proposed system, the fitness must be based on the profit and frequency of the pattern. The fitness F_i of the chromosome can be calculated as follows

$$F_i = \left(1 - \frac{1}{gu_i} \right) \quad (4)$$

In this eqn (4), the gu_i represents the utility assigned for the i^{th} chromosome and it can be calculated as follows

$$gu_i = \frac{1}{n} \sum_{i=1}^n (fq_i \times pf_i) \quad (5)$$

$$pf_i = \sum_{j=1}^{|pr|} (Ut_j) \quad (6)$$

where, fq_i represents the frequency of the i^{th} gene, pf_i represents the profit of the i^{th} chromosome and Ut_j represents the utility of j^{th} gene.

Crossover: After establishing the selected individuals, to create new individuals a process of “crossover” is carried out between the individuals. The existing individuals are called the “Parents”, and the new individuals are called the “Children”. Crossover operator is designed in such a way that it transfers genetic material from one generation to another. Validity and context insensitivity are the

major concerns with this operator. It is applied with probability P_x , after the pairs are chosen for breeding. In crossover, the genes of the parent chromosomes that are at the right side of the crossover point are interchanged between the parents to obtain the children chromosomes. Hence, from the genetic operation N_p children chromosomes are obtained from N_p parent chromosomes, and the obtained children chromosomes are represented as follows

$$X_{child}^{(k)} = \left[x_0^{(k)} \ x_1^{(k)} \ x_2^{(k)} \ \dots \ x_{N_{Tx}-1}^{(k)} \right], \quad N_p \leq k \leq 2n_c - 1 \tag{7}$$

In the next step the obtained N_p chromosomes are subjected to mutation.

Mutation: In addition to selection and crossover, a small percentage of individuals undergo mutation. Similar to crossover, mutation takes place with relation to some probability P_m which is usually quite low. An element of an individual’s chromosome is randomly selected by employing this probability and altered. In binary coding, this simply means changing a “0” to a “1” or a “1” to a “0”. Mutation is another way of creating diversity of population and examining parts of the solution space that may otherwise have been unnoticed. With a set probability P_m mutation is implemented to the novel chromosomes. Mutations induce a change in the individual genetic representation according to certain probabilistic rules.

Chromosome selection: A completely fresh population of $2N_p$ chromosomes is attained after the process of crossover and mutation. The fitness will be calculated for this new population. Among these $2N_p$ chromosomes, the N_p chromosomes which have the utmost fitness are chosen as the new parent chromosomes. The similar process of crossover, mutation, fitness evaluation and selection are repeated until the iteration value reaches the utmost iteration limit n_c . After this, the process is terminated and we have N_p chromosomes which represent the fit individuals. Thus, we obtain N_p optimized fuzzy rules which have maximum utility and profit. The complete process of our optimization technique is illustrated in the following figure.

In fig.2, the most fit individual represents the optimum chromosome that is obtained. A set of sequential patterns is available in all chromosomes. On the basis of the fitness value and profit these sequential patterns, the optimum sequential pattern is chosen. The optimum chromosome has utmost utility. If we have huge amount of sequential pattern at the end of mining means, our proposed system will effectively chose the sequential patterns which has more profit. So, instead of choosing all the patterns, the user can only choose the profitable patterns. This will reduce the computational cost and time. If we can focus on only those sequential patterns interesting to users, we may be able to save a lot of computation cost by those uninteresting patterns.

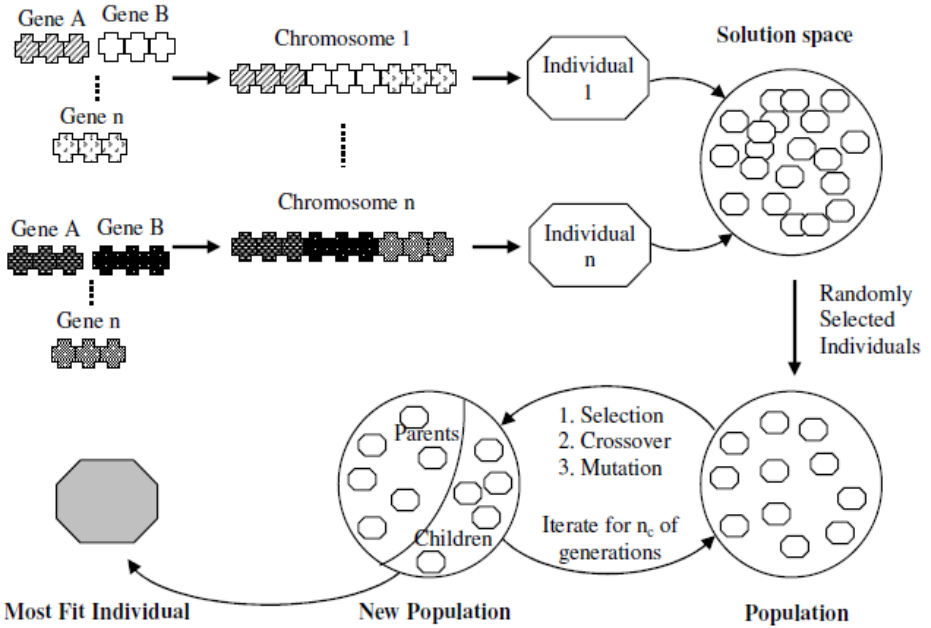


Fig. 2. Process of Optimization Technique

5 Results and Discussion

The data mining system, which is proposed in this paper, was implemented in the working platform of JAVA (version JDK 1.6). Tracking of goods in warehouses was evaluated for the proposed system that employs RFID application. Here, it is assumed that the warehouse contains six stationery goods; they are Pencil, Pen, Notebook, Diary, Paper clips, Memo holder and so $N_{cl} = 6$. The RFID tags are attached with every product and hence each product has its own EPC code. The warehouse is estimated to have 200 products/per class ($N_T = 200$) and eight RFID readers (i.e. $N_R = 8$). The I-dataset is generated from the warehoused data. Then, with a least support $\sup_{\min} = 1$ the sequential patterns are generated.

The fuzzy rules are generated with the help of the sequential patterns. After the generation of the fuzzy rules, the optimal rule based on their utility is generated by employing GA. For this optimization technique, we use $P_x = 0.4$ as the crossover rate and $P_m = 0.2$ as the mutation rate. The top five optimum chromosomes obtained are depicted in **Table 1**. In this table, chromosome represents the fuzzy rules. This table represents the products, rules and their corresponding profit. The first chromosome is the optimum one that has maximum profit and fitness values.

Table 1. Representation of top five chromosomes and its corresponding fitness and profit

Chromosome No.	Product N_{cl}	Rule	Profit (in Rs.)	Fitness (in %)
1	2 \cap 5	r1 , r3 \rightarrow r2	50	97.803
	2 \cap 5	r5 , r1 \rightarrow r6 , r8	50	
	2 \cap 4	r3 \rightarrow r5 , r1	42	
	2 \cap 4	r3 , r1 \rightarrow r4 , r6	42	
	3 \cap 5	r7 , r6 \rightarrow r3 , r2	45	
2	2 \cap 3	r8 \rightarrow r7 , r2	35	97.658
	2 \cap 5	r5 , r1 \rightarrow r6 , r8	50	
	4 \cap 5	r7 \rightarrow r6 , r1	52	
	3 \cap 4	r7 \rightarrow r5 , r3	37	
	3 \cap 5	r7 , r6 \rightarrow r3 , r2	45	
3	5	r2 , r4 , r7 , r6 \rightarrow r3	30	97.632
	2 \cap 5	r5 , r1 \rightarrow r6 , r8	50	
	4 \cap 5	r7 \rightarrow r6 , r1	52	
	2 \cap 4	r3 , r1 \rightarrow r4 , r6	42	
	3 \cap 5	r7 , r6 \rightarrow r3 , r2	45	
4	2 \cap 3	r8 \rightarrow r7 , r2	35	97.628
	2 \cap 5	r5 , r1 \rightarrow r6 , r8	50	
	2 \cap 4	r3 \rightarrow r5 , r1	42	
	2 \cap 4	r3 , r1 \rightarrow r4 , r6	42	
	3 \cap 5	r7 , r6 \rightarrow r3 , r2	45	
5	2 \cap 3	r8 \rightarrow r7 , r2	35	97.567
	2 \cap 5	r5 , r1 \rightarrow r6 , r8	50	
	2 \cap 4	r3 \rightarrow r5 , r1	42	
	3 \cap 4	r7 \rightarrow r5 , r3	37	
	3 \cap 5	r7 , r6 \rightarrow r3 , r2	45	

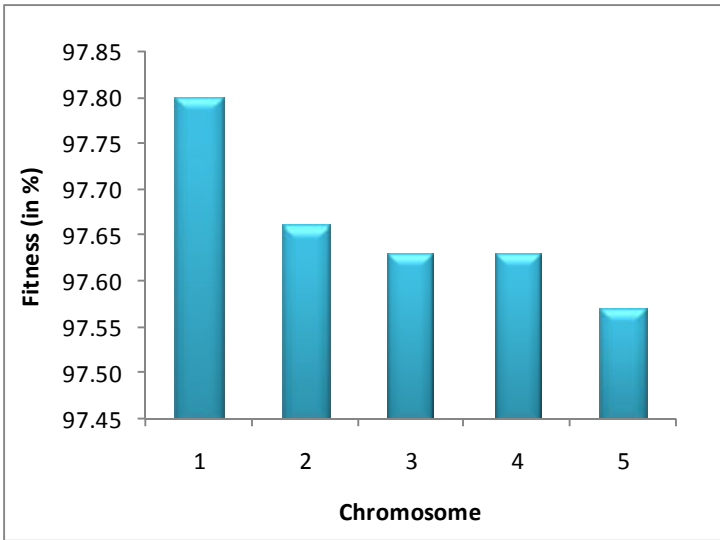


Fig. 3. Flow of fitness values for the top five chromosomes

Fig. 3 depicts the optimum fitness value. This figure illustrates the fitness values of the top five chromosomes. The first chromosome is the optimum chromosome with utmost fitness value. The profit of the top five chromosomes are depicted in fig. 4 and the first chromosome gives the maximum profit.

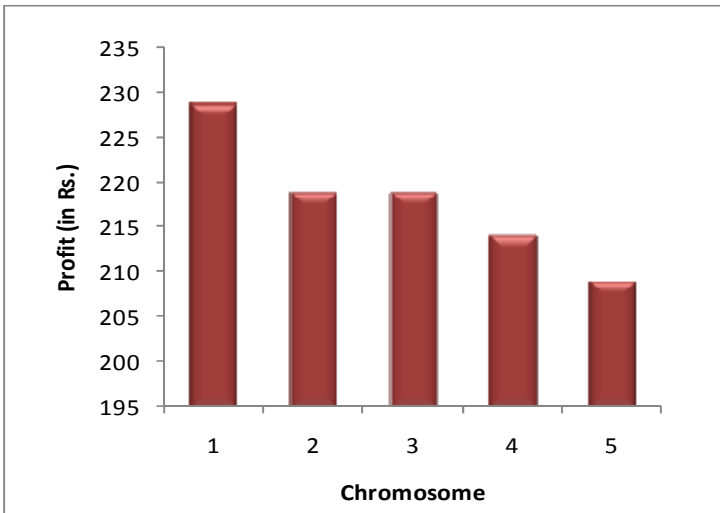


Fig. 4. Flow of Profit for the top five chromosomes

Fig. 5 illustrates the flow of profit and fitness values of the chromosomes after completing 50 iterations. The optimum value for both fitness and profit is reached on completion of the 50th iteration.

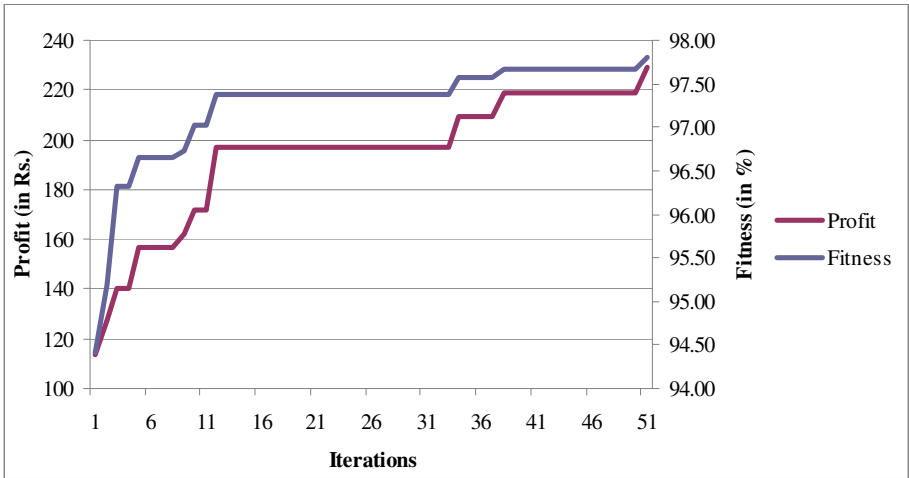


Fig. 5. Flow of profit and fitness values for the top five chromosomes

Analysis of chromosomes

By applying GA to the fuzzy rules, the optimum rule with maximum profit and fitness is identified. The optimum chromosome has the fitness $F_i = 0.97803$ and profit $pf_i = Rs.229.0$. So we analyze the optimum chromosome by comparing it with other chromosomes to check whether it is truly optimum or not. This process is illustrated in Fig.6 and Fig.7.

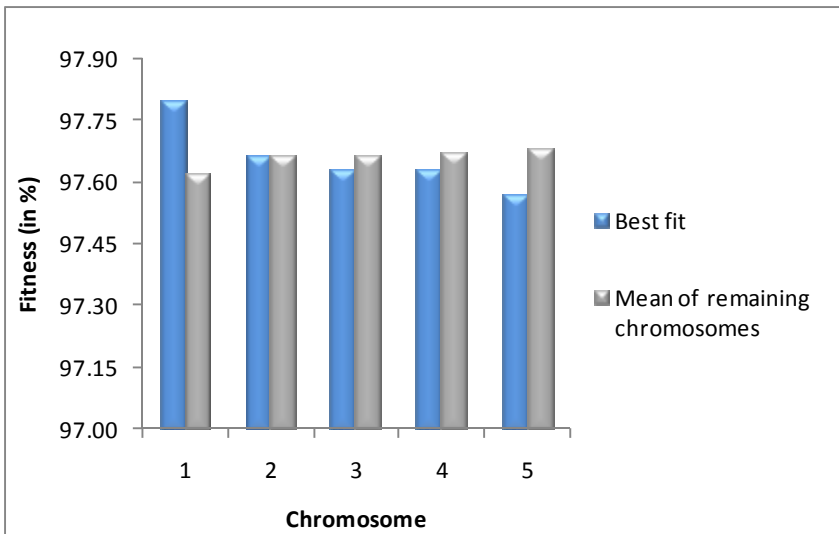


Fig. 6. Comparison of fitness values between the chromosomes

In Fig.6, the fitness value of each of the top five chromosomes are analyzed with mean fitness values of the remaining four chromosomes. When the first chromosome is selected as the best fit, its fitness value is considerably greater than the mean fitness value of the remaining four chromosomes. But, when the second chromosome is selected as the best fit, its fitness value is slightly less than the mean fitness value of the remaining four chromosomes. This is so because the first chromosome is also included in the remaining chromosomes part. Thus it is evident that the proposed method selected optimum chromosome (here. first chromosome) for which the fitness value dominates in the result is truly the optimum chromosome.

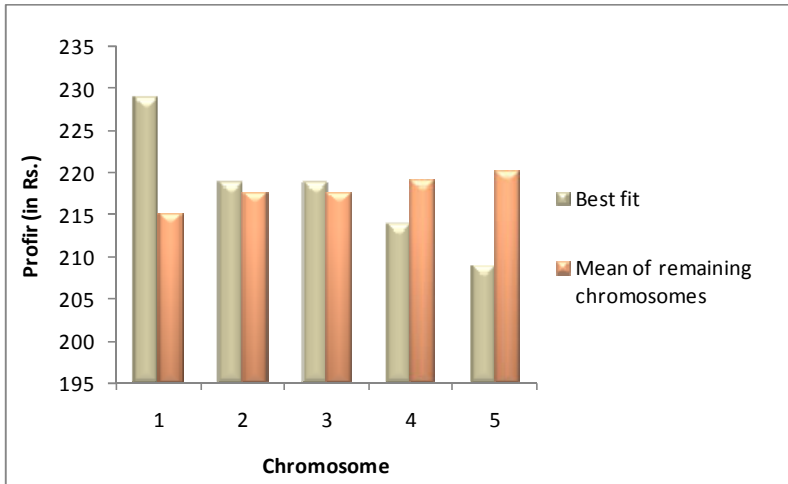


Fig. 7. Analysis of profit values between the chromosomes

Similarly, the profits of the top five chromosomes are also analyzed. In fig.7, the first one represents the optimum chromosome with maximum profit which is compared with the mean profit value of the remaining four chromosomes. But for the second one, the profit of the best fit is slightly higher than the mean of remaining chromosomes. This is due to the inclusion of the profit of the optimum chromosome in calculating the mean of the remaining chromosomes. So it dominates the profit results. Likewise, for the other three chromosomes, the optimum chromosome dominates the results. So the profit of best fit is less than the mean of remaining chromosomes. From this analysis, it is clear that the best fit is obtained with the optimum chromosome.

Comparison of Results

The proposed optimization technique is used to generate the optimum sequential pattern based on their utility. Here we compare the results of the optimum chromosome with that of all other remaining chromosomes. Table 2 represents the fitness and profit value of the best chromosome and the mean of the remaining chromosomes. Here it is evident that the best chromosome has the highest fitness and profit values.

Table 2. Comparison of fitness and profit between best fit and other chromosomes

Performance	Best Chromosome	Mean of Remaining Chromosomes
Fitness (in %)	97.8031746	96.6249912
Profit (in Rs.)	229.0	196.1579

6 Conclusion

In this paper, we have presented a data mining system for mining information related to the movement of tags attached to warehoused goods from such type of data stored in huge databases. The proposed mining system mines knowledge from the warehoused data by first generating I-dataset, followed by mining of sequential patterns and finally generating fuzzy rules from the mined sequential patterns. After that, on the basis of their assigned utility, the sequential patterns are optimized using GA. The fuzzy score obtained as the output of the system indicates the profit associated with the type of tag movement corresponding to the optimized fuzzy rule. Given a part of the tag (indirectly it refers to a product) movement, the fuzzy rules identifies the remaining path of the tag (product). In this manner, diverse length combinations of tags are taken into consideration and their movements are understood. The movements were considered only for some important tags and combinations and not for all tags and their combinations. From the implementation results and comparative analysis, it is evident that our proposed system efficiently identifies the optimum sequential pattern. So, with the help of the presented optimized data mining system, tracking of goods in large warehouses can be performed efficiently. As we only concentrated on the optimized sequential patterns the cost of mining the sequential patterns is very low. The extracted information would be useful for warehouse management.

References

1. Li, B., Shasha, D.: Free Parallel Data Mining. *ACM SIGMOD Record* 27(2), 541–543 (1998)
2. Anand, S.S., Bell, D.A., Hughes, J.G.: EDM: A general framework for data mining based on evidence theory. *Data and Knowledge Engineering* 18(3), 189–223 (1996)
3. Agrawal, R., Imielinsk, T., Swami, A.: Database Mining: A Performance Perspective. *IEEE Transaction Knowledge and Data Engineering* 5(6), 914–925 (1993)
4. Chen, S.Y., Liu, X.: Data mining from 1994 to 2004: an application-oriented review. *International Journal of Business Intelligence and Data Mining* 1(1), 4–11 (2005)
5. Singh, D.R.Y., Chauhan, A.S.: Neural Networks In Data Mining. *Journal of Theoretical and Applied Information Technology* 5(6), 36–42 (2009)
6. Roberts, C.M.: Radio frequency identification (RFID). *Computers & Security* 25, 18–26 (2006)
7. Aboalsamh, H.A.: A novel Boolean algebraic framework for association and pattern mining. *WSEAS Transactions on Computers* 7(8), 1352–1361 (2008)

8. Sathiyamoorthi, V., Bhaskaran, V.M.: Data Mining for Intelligent Enterprise Resource Planning System. *International Journal of Recent Trends in Engineering* 2(3), 1–5 (2009)
9. Ranjan, J., Bhatnagar, V.: A Review of Data Mining Tools In Customer Relationship Management. *Journal of Knowledge Management Practice* 9(1) (2008)
10. Shaw, M.J., Subramaniam, C.S., Tan, G.W., Welge, M.E.: Knowledge management and data mining for marketing. *Decision support systems* 31(1), 127–137 (2001)
11. Sabbaghi, A., Vaidyanathan, G.: Effectiveness and Efficiency of RFID technology in Supply Chain Management: Strategic values and Challenges. *Journal of Theoretical and Applied Electronic Commerce Research* 3(2), 71–81 (2008) ISSN 0718–1876
12. Asif, Z., Mandviwalla, M.: Integrating the supply chain with RFID: a technical and business analysis. *Communications of the Association for Information Systems* 15(24), 393–427 (2005)
13. Pei, J., Han, J., Asl, B.M., Wang, J., Pinto, H., Chen, Q., Dayal, U., Hsum, M.C.: Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach. *IEEE Transactions on Knowledge and Data Engineering* 16(10), 1–17 (2004)
14. Chen, M.S., Han, J., Yu, P.S.: Data mining: an overview from a database perspective. *IEEE Transactions on Knowledge and Data Engineering* 8(6), 866–883 (1996)
15. Chen, Y.L., Hu, Y.H.: Constraint-based sequential pattern mining: The consideration of recency and compactness. *Decision Support Systems* 42, 1203–1215 (2006)
16. Jea, K.F., Lin, K.C., Liao, I.E.: Mining hybrid sequential patterns by hierarchical mining technique. *International Journal of Innovative Computing, Information and Control* 5(8) (2009)
17. Saputra, D., Rambli, D.R.A., Foong, O.M.: Mining Sequential Patterns Using I-PrefixSpan. *International Journal of Computer Science and Engineering* 2(2), 49–554 (2008)
18. Hu, J., Mojsilovic, A.: High-utility pattern mining: A method for discovery of high-utility item sets. *Pattern Recognition* 40, 3317–3324 (2007)
19. Pei, J., Han, J., Wang, W.: Constraint-based sequential pattern mining: The pattern-growth methods. *Journal of Intelligent Information Systems* 28(2), 133–160 (2007)
20. Sakurai, S., Kitahara, Y., Orihara, R.: A Sequential Pattern Mining Method based on Sequential Interestingness. *International Journal of Computational Intelligence* 4(4), 252–260 (2008)
21. Exarchos, T.P., Tsiouras, M.G., Papaloukas, C., Fotiadis, D.I.: A two-stage methodology for sequence classification based on sequential pattern mining and optimization. *Data & Knowledge Engineering* 66, 467–487 (2008)
22. Shankar, S., Purusothaman, T.: Utility Sentient Frequent Itemset Mining and Association Rule Mining: A Literature Survey and Comparative Study. *International Journal of Soft Computing Applications* 10(4), 81–95 (2009)
23. Ykhlef, M., ElGibreen, H.: Mining Sequential Patterns Using Hybrid Evolutionary Algorithm. *World Academy of Science, Engineering and Technology* 60, 863–870 (2009)
24. Pillai, J., Vyas, O.P.: Overview of Itemset Utility Mining and its Applications. *International Journal of Computer Applications* 5(11), 9–13 (2010)
25. Sedighzadeh, M., Rezazadeh, A.: Using Genetic Algorithm for Distributed Generation Allocation to Reduce Losses and Improve Voltage Profile. *World Academy of Science, Engineering and Technology* 37 (2008)
26. Radhakrishnan, P., Prasad, V.M., Gopalan, M.R.: Optimizing Inventory Using Genetic Algorithm for Efficient Supply Chain Management. *Journal of Computer Science* 5(3), 233–241 (2009)

27. Al-Maqaleh, B.M., Bharadwaj, K.K.: Genetic Programming Approach to Hierarchical Production Rule Discovery. *World Academy of Science, Engineering and Technology* 11, 43–46 (2005)
28. Mantere, T.: A Min-Max Genetic Algorithm with Alternating Multiple Sorting for Solving Constrained Problems. In: *Proceedings of the Ninth Scandinavian Conference on Artificial Intelligence* (2006)
29. Diaz-Gomez, P.A., Hougen, D.F.: Improved Off-Line Intrusion Detection Using A Genetic Algorithm. In: *Proceedings of the Seventh International Conference on Enterprise Information Systems*, Miami, USA, May 25-28, pp. 66–73 (2005)
30. Reddy, S.R.: Selection of RTOS for an Efficient Design of Embedded Systems. *International Journal of Computer Science and Network Security* 6(6), 29–37 (2006)
31. Schenk, S., Hanke, K.: Combining Genetic Algorithms With Imperfect And Subdivided Features For The Automatic Registration Of Point Clouds (GAREG-ISF). In: *Proceedings of the 3rd ISPRS International Workshop*, vol. 38 (2009)
32. Korejo, I., Yang, S., Li, C.: A Comparative Study of Adaptive Mutation Operators for Genetic Algorithms. In: *Proceedings of the 8th Metaheuristic International Conference*, July 13-16 (2009)
33. Sewell, M., Samarabandu, J., Rodrigo, R., McIsaac, K.: The Rank-scaled Mutation Rate for Genetic Algorithms. *International Journal of Information Technology* 3(1) (2006)
34. Bankovic, Z., Moya, J.M., Araujo, A., Bojanic, S., Taladriz, O.N.: A Genetic Algorithm-based Solution for Intrusion Detection. *Journal of Information Assurance and Security* 4, 192–199 (2009)

SYEDWSIM: A Web Based Simulator for Grid Workload Analysis

Syed Nasir Mehmood Shah, Ahmad Kamil Bin Mahmood, and Alan Oxley

Department of Computer and Information Sciences
Universiti Teknologi PETRONAS,
Bandar Seri Iskandar, Tronoh,
31750, Perak, Malaysia

nasirsyed.utp@gmail.com, {kamilmh, alanoxley}@petronas.com.my

Abstract. Grid computing is becoming the most demanding platform for solving large-scale scientific problems. Grid scheduling is the core component of a Grid infrastructure. Grid scheduling plays a key role in the efficient and effective execution of Grid jobs. In this context, understanding the characteristics of real Grid workloads is a critical step for improving the quality of an existing Grid scheduler, and in guiding the design of new scheduling solutions. Towards this goal, in this paper we present our developed web based simulator for the statistical analysis of Grid workload traces. Our web based simulator provides a comprehensive characterization of the real workload traces. Metrics that we characterize include system utilization, job arrival rate and inter-arrival time, job size (degree of parallelism), job runtime, data correlation and Fourier analysis. Our paper provides a realistic basis for experiments in resource management and evaluations of different job scheduling algorithms in Grid computing.

Keywords: Distributed systems; Cluster; Grid computing; Grid scheduling; workload modeling; performance evaluation; Simulation; parallel processing.

1 Introduction

Grid computing can be viewed as a mainstream technology for large-scale resource sharing [1]. Grid computing increases a system's computing capability and the tendency has been to use it to solve complex and large-scale scientific problems using geographically dispersed computing resources. A large number of complex and large-scale scientific issues cannot be solved by using a traditional network or super computer; that is why research and development of Grid computing has been making progress gradually. A Grid computing system connects available computing resources, such as computers, applications, and storages devices, to networks for high performance computing and reduces system execution time [2, 3].

Grid scheduling is a process of ordering tasks on compute resources and ordering communication between them. It is also known as the allocation of computation and communication over time [4]. Grid scheduling is a core component of a Grid and is

responsible for efficient and effective utilization of heterogeneous and distributed resources.

New Grid scheduling components cannot be designed without a good understanding of how today's Grids are used, and of their performance. Also, existing Grid schedulers cannot be evaluated without understanding the characteristics of real Grid workloads [5, 6, 7, 8]. In this perspective, the study of the nature of real Grid workloads is a vital step for improving the quality of existing Grid schedulers.

Different researchers use different programs for analysis of workload traces. A number of approaches have been developed for statistical analysis of workload traces. In this paper, we propose an alternative web based approach to simulating such environments that uses resource utilization traces from real deployments. We designed and developed a web based simulator to perform the statistical analysis of Grid workload traces. We can perform a detailed analysis on any real Grid workload trace to quantify the performance of the Grid systems from different perspectives, e.g. users, groups and individual jobs characteristics. Our simulator takes the real trace as input in the GWF (Grid Workload Format) format as detailed in [9] and produces a variety of graphs to analyze the characteristic of workloads. In this paper, we reproduced the analysis of two well known traces from two real Grid environments, namely LCG [9] and DAS-2 [10]. LCG is scientific Grid while DAS-2 is a research Grid.

We discuss our proposed simulation technique and present results from our developed simulator (i.e., SyedWSim). We have validated our computed results in using the simulator by comparing with the results available [9, 10, 11]. Our simulation results show that this web based simulator would facilitate the research community in studying the nature of different Grids and give guidance in the evaluation of scheduling algorithms.

This research is motivated by our desire to improve the quality of our own work in the performance evaluation of scheduling algorithms and to help facilitate the comparability of results obtained in the scheduling community. We wished to develop a web based simulator for statistical analysis of real workload traces for use in the Grid resource management community.

The structure of the paper will now be described. Section 2 is a literature review of Grid workload archives. Section 3 presents the design and development of the web based simulator. Section 4 is about the GUI of the web based simulator. Section 5 describes practical applications of statistical theory. Section 6 is about the statistical analysis of real workload traces using our developed simulator and section 7 concludes the paper.

2 Related Research

A Grid is a high performance computational system which consists of a large number of distributed and heterogeneous resources. Grid computing enables sharing, selection and aggregation of resources to solve the complex large-scale problems in science, engineering and commerce. Scientific applications usually consist of numerous jobs that are processed and generate large datasets. Processing complex scientific applications in a Grid imposes many challenges due to the large number of jobs, file

transfers and the storage needed to process them. The scheduling of jobs focuses on mapping and managing the execution of tasks on shared resources [6].

[12] states that Computational Grids have been “inspired by the electrical power Grid’s pervasiveness, ease of use and reliability”. A Computational Grid is a type of parallel and distributed system of computers. Grid computing enables the sharing, selection and aggregation of a wide variety of resources. Grid resources are geographically distributed and owned by different organizations. They are used for solving large-scale problems in science, engineering and commerce. The ‘nodes’ of the Grid may be individual computers but are more likely to be computer sites. A Grid may consist of resources owned by a number of different organizations, within which sharing arrangements have been established. Most Grids use the idle time of thousands or millions of computers throughout the world.

Grid scheduling is defined as the process for making the scheduling decisions involving resources over multiple administrative domains. This process also includes searching multiple administrative domains to assign a job to a single machine or scheduling a job’s tasks to multiple resources at a single site or multiple sites [13].

There are three main phases of Grid scheduling. Phase one is resource discovery, which provides a list of available resources. Phase two is resource allocation, which involves the selection of feasible resources and the mapping of jobs to resources. In the second phase, the selection of the best match of jobs to resources is an NP-complete problem. The third phase is job execution, which includes file staging and cleanup [14].

Two fundamental issues have to be considered for the performance evaluation of new Grid scheduling algorithms. Firstly, representative workload traces are required to produce dependable results [15]. Secondly, a good testing environment should be set up, most commonly through simulations. A standard workload should be used as a benchmark for evaluating scheduling algorithms [16, 17].

The workload plays a significant role in experimental performance evaluation of computer systems. Workload characterization is important to understand the system performance. There is a need to develop workload models for evaluating different Grid scheduling strategies [18, 19]. Researchers have put a lot of effort into real workload collection [9], analysis [20, 21, 22], and modeling [23, 24]. There is a need for a web simulator to facilitate researchers in performing the statistical analysis of the Grid workload traces; and evaluating and improving the performance of Grid schedulers. The aim of this paper is to present a new web based simulator for statistical analysis of real workload traces. This simulator would facilitate researchers in studying the nature of different Grids and evaluating the performance of Grid scheduling algorithms.

3 Design and Development of Web Based Simulator

We designed and developed a web based simulator ‘*SyedWSIM*’ using Java. Our web based simulator needs the input workload to be in the GWF format; the format is described by [9]. A work flow diagram of *SyedWSim* is shown in Figure 1. It takes real workload traces (also called resource utilization traces), workload percentage and interval size as input. It models the impact of different scheduling policies on Grid

performance under various loads and policies. It converts a source workload file into a relational data structure at runtime for analysis purposes.

Our web based simulator uses a timing model to depict the system’s dynamic behaviour. The model makes use of the jobs’ time units, as given in the workload file. The basic operation of the simulator will now be described. The main simulator loop operates on a timer that records time in the simulated system. When reading the input trace file, the simulator uses this timer to access workload data at the corresponding time stamp associated with it. Our web based simulator models to the users’ characteristics, groups’ characteristics and Grid jobs’ characteristics.

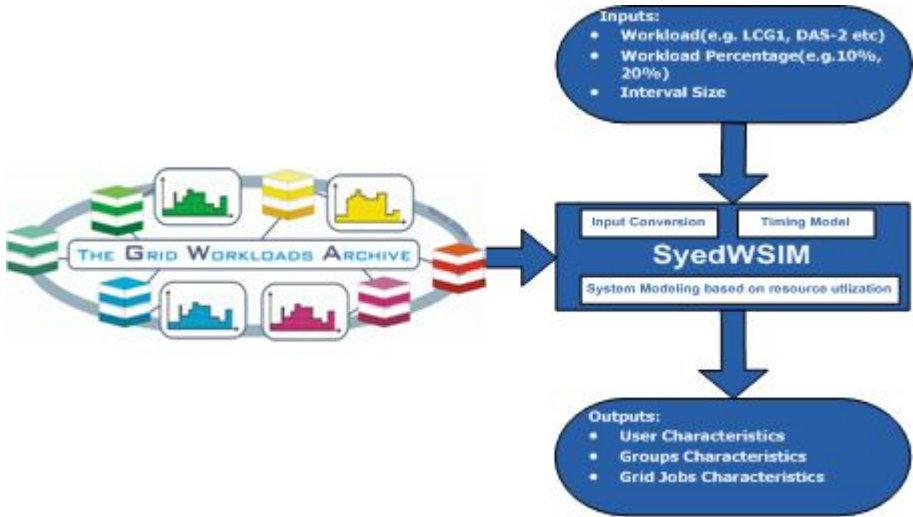


Fig. 1. Work flow diagram of SyedWSim

Our developed package consists of the following components, as represented in the class diagram shown in Figure 2. Now, each class component will be described.

3.1 CV2DChartPanel.Java

This class includes graph drawing functionality for all features of the web simulator related to the Grid users, virtual organizations and jobs. It interacts with all other classes of the simulator and produces output depicting the nature of the workload under analysis. This class includes 13 methods (namely AddVisualizationMethod1, AddVisualizationMethod2, etc.) as shown in Figure 2, and each method corresponds to each feature available with the web simulator (i.e. user & total jobs, user & log(total jobs), etc.) as highlighted by ‘Options’ in Figure 3.

3.2 CVApplet.Java

It is a web based applet file which executes on a client browser. It includes a visualization class which extends the functionality of the Java Applet class. It also includes all visual components and provides all visual interfaces to interact with the web simulator.

3.3 CVToolBar.Java

This class provides a toolbar which includes a list of options allowing the user to experiment with the web based simulator for workload analysis.

3.4 Dataset.Java

This class file is used to download the dataset from the original text file, which is in GWF format, and to construct the dynamic data structure at runtime.

We also used open source code in the development of the web based simulation. This will now be described.

3.5 Package edu.emory.mathcs.jtransforms.fft

We used the ‘JTransforms’ package [25] to apply the Fast Fourier Transform (FFT) to the values obtained by the autocorrelation function.

3.6 Package info.monitorenter.gui.chart

We used a ‘GUI’ package, entitled JChart2D [26], to produce the 2D Chart. We used JChart2D for displaying the data contained in an ITrace2D. JChart2D inherits a number of features from javax.swing.JPanel. The package is for use where we are more interested in showing the results than the cosmetics of the output.

3.7 Package org.apache.commons.math

We used an open source math package, entitled Commons Math [27], for mathematical and statistical calculations.

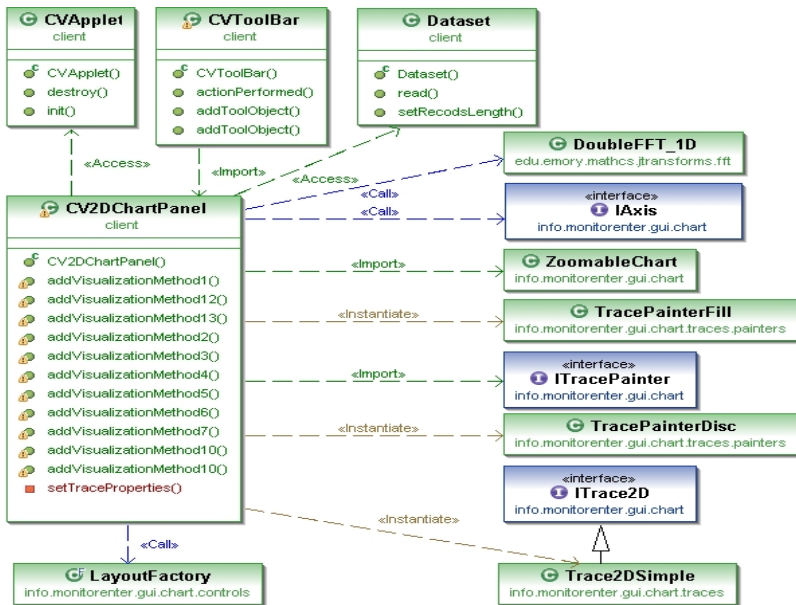


Fig. 2. Class Diagram of SyedWSim

4 GUI of SyedWSim

The Graphical User Interface of our developed simulator (i.e., SyedWSim) is shown in Figure 3.

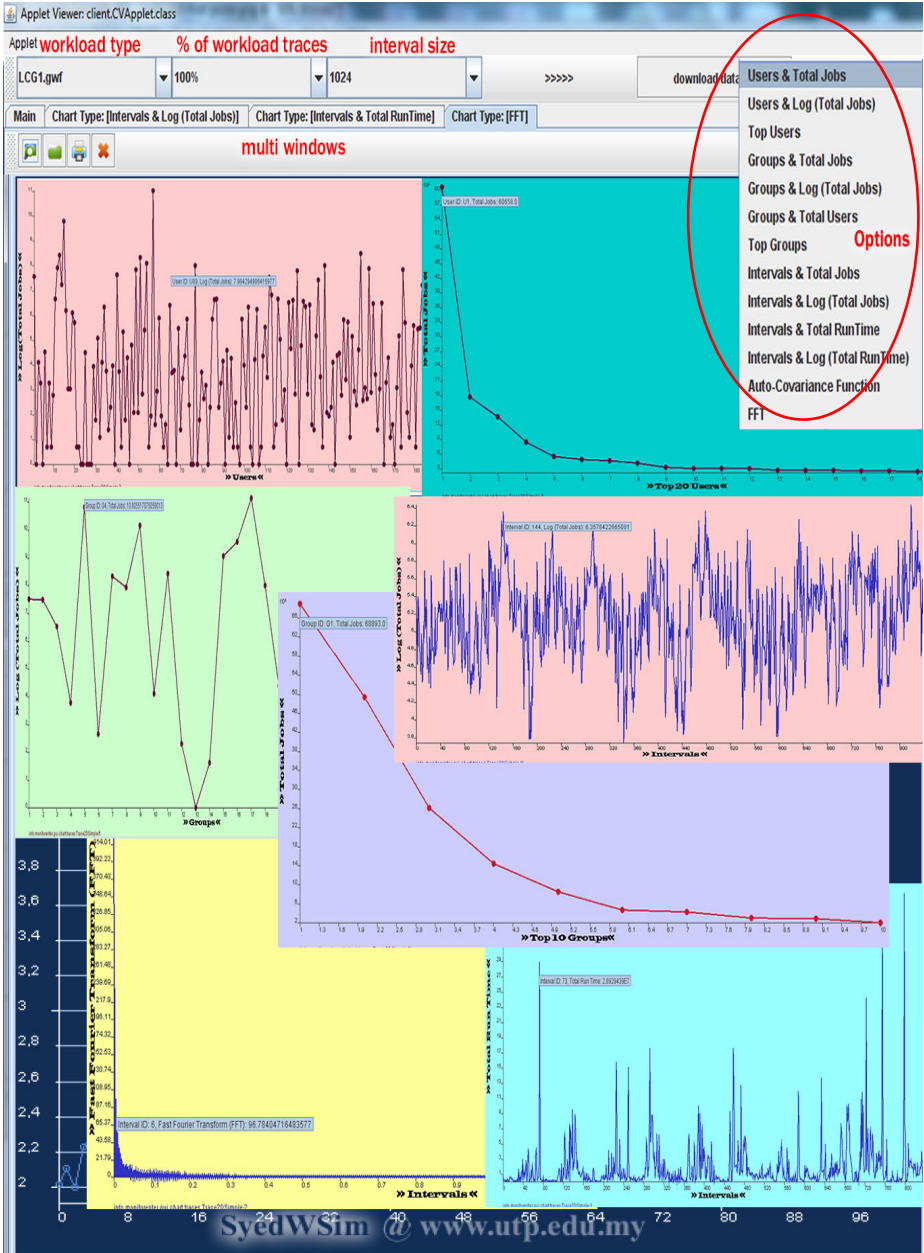


Fig. 3. GUI of SyedWSim

5 Practical Application of Statistical Theory

The web based simulator is an application for statistical analysis. One technique is autocorrelation. Autocorrelation is the cross-correlation of data with itself. It is a function to find the similarity between a list of observations, and the same list offset by a certain ‘lag.’ It is a mathematical tool to find repeating patterns. In our simulator the list comprises job inter-arrival times.

Another technique used in statistical analyses is Fourier analysis. We used the JTransforms package [25] for Fourier analysis.

6 Statistical Analysis of Workloads Using SyedWSim

In [11], a comprehensive statistical analysis has been carried out for a variety of workload traces on clusters and Grids. We reproduced the graphs of [11] to study the behavior of the dynamic nature of workloads LCG1 and DAS-2, using our SyedWSim. The total numbers of jobs in LCG1 and DAS-2 are 188041 and 1124772, respectively. We looked at the number of jobs arriving in each 64 second period (i.e. the interval size). The number of jobs arriving in a particular period is its ‘job count’. In our experiments, we drop trace job entries that have a negative runtime or a negative number of allocated processors.

The input format (in GWF format) of the LCG1 trace is shown in Table 1. Some columns of the trace are filled with ‘-1’. This means that the data is not given. The data that is given includes that shown in Table 1. Only the first three of the 188,041 jobs are shown. Each job is submitted by a user. There are different groups of users. As an example, the first job in the trace, job 1, was submitted by user U1, who is a member of group G1.

Table 1. Trace LCG1

1	2	4	5	12	13	16	17	18
Job ID	Submit time	Run Time	NProcs	User ID	Group ID	Partition ID	Orig Site ID	Last Run Site ID
1	1132444805	83	1	U1	G1	1	SWF	SWF
2	1132444808	3611	1	U2	G2	2	SWF	SWF
3	1132444817	205	1	U1	G1	3	SWF	SWF
...								

Figure 4 shows the user input for the workload traces LCG1 and DAS-2 respectively. Figure 5 shows the number of jobs per user for LCG1 and DAS-2. Figure 6 shows the magnitudes of the top 15 users. User ‘U1’ is the topmost user in LCG1; who submitted 60658 jobs to the system for execution. Tooltip shows the relevant statistics at the mouse cursor position for each diagram. Figure 7 shows the number of jobs per group, whilst figure 8 shows the number of users for each group.

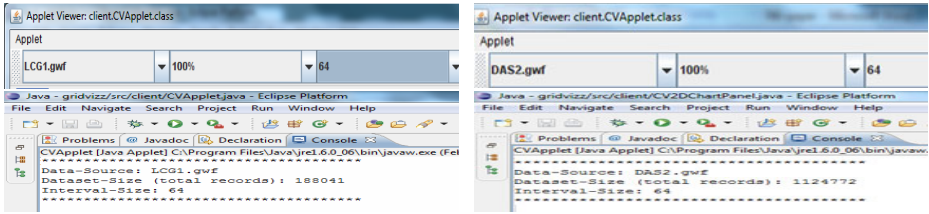


Fig. 4. The user input for LCG1 and DAS-2

6.1 Users Characteristics

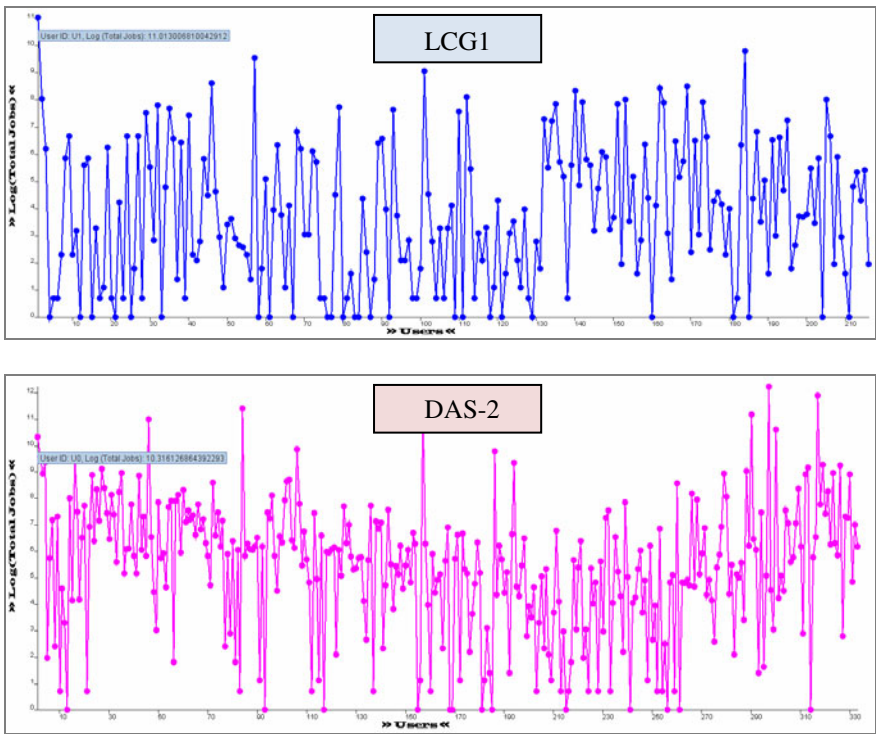


Fig. 5. The user jobs for LCG1 and DAS-2

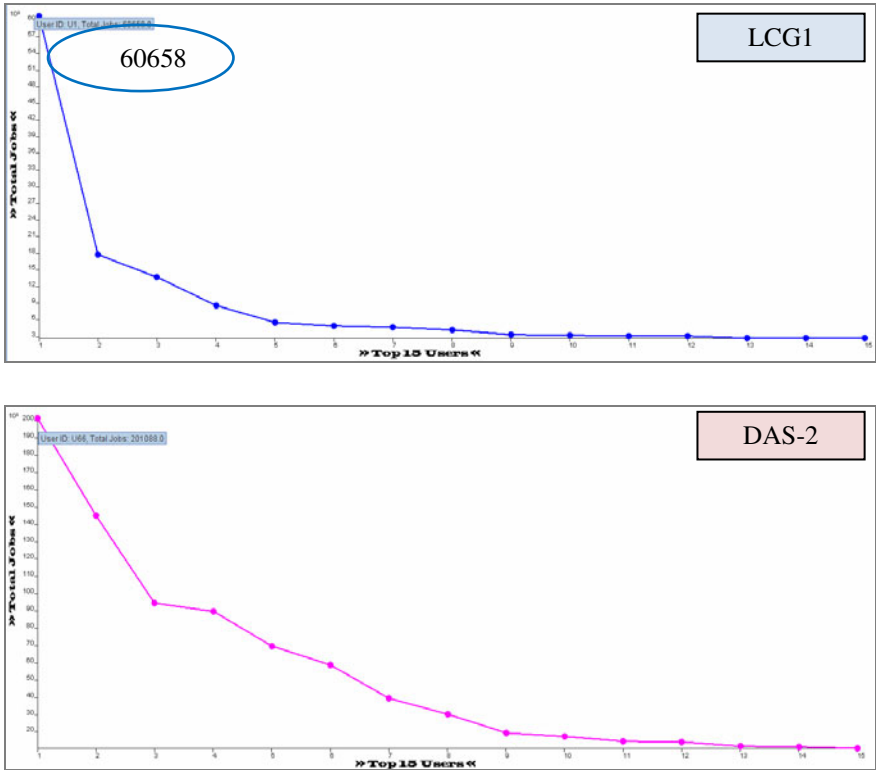


Fig. 6. Top 15 users for LCG1 and DAS-2

6.2 Groups Characteristics

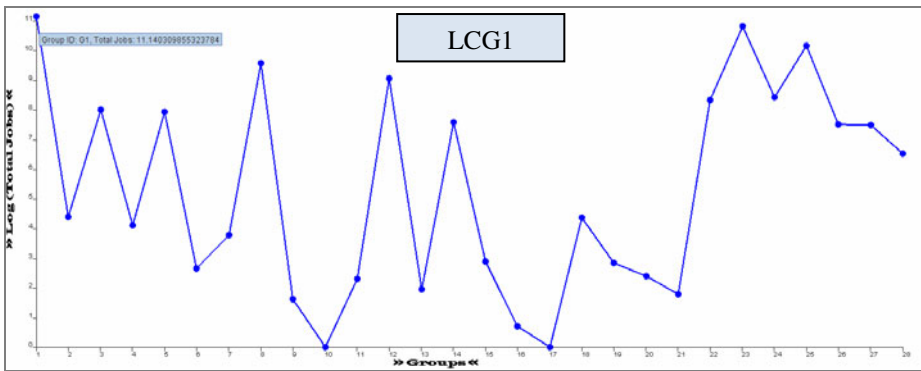


Fig. 7. The Group jobs for LCG1 and DAS-2

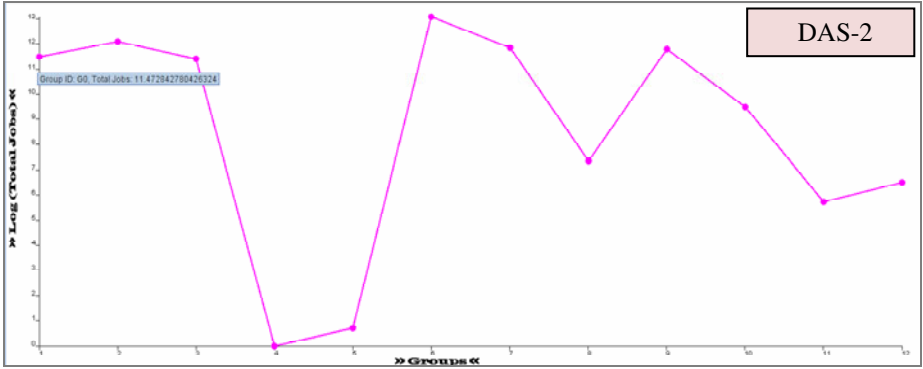


Fig. 7. (continued)

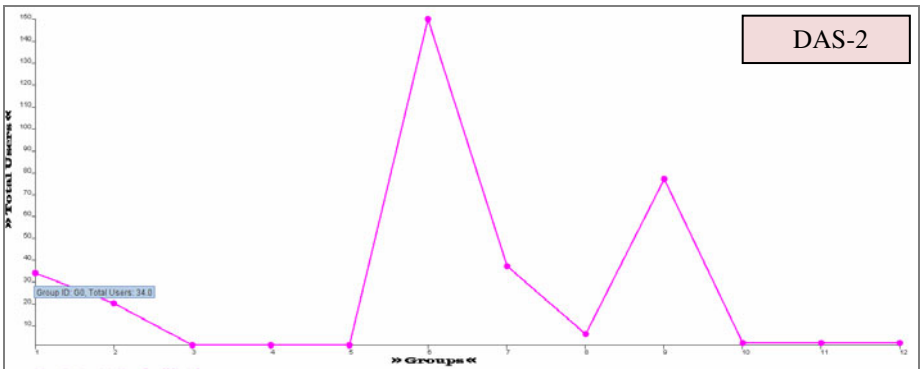
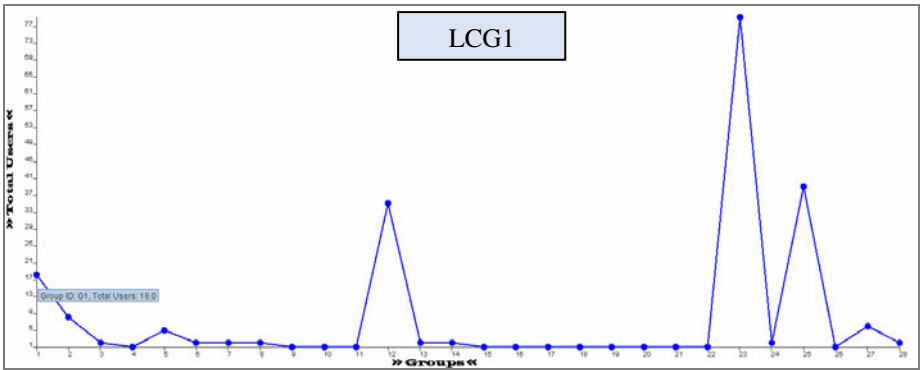


Fig. 8. Groups vs Number of Users for LCG1 and DAS-2

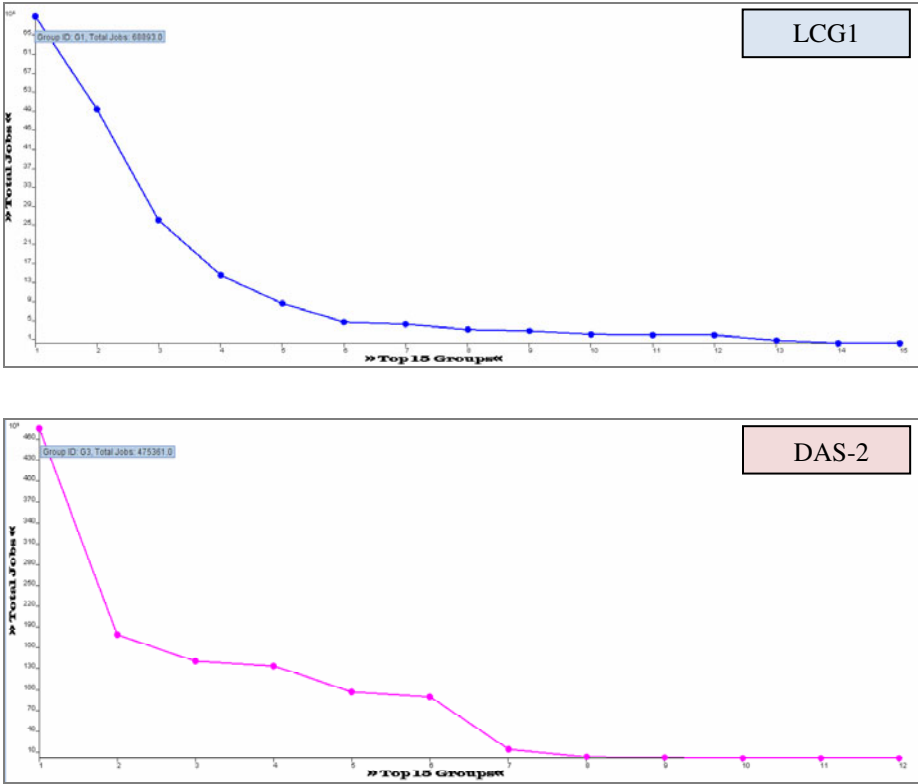


Fig. 9. Top 15 Groups for LCG1 and DAS-2

6.3 Grid Jobs Characteristics

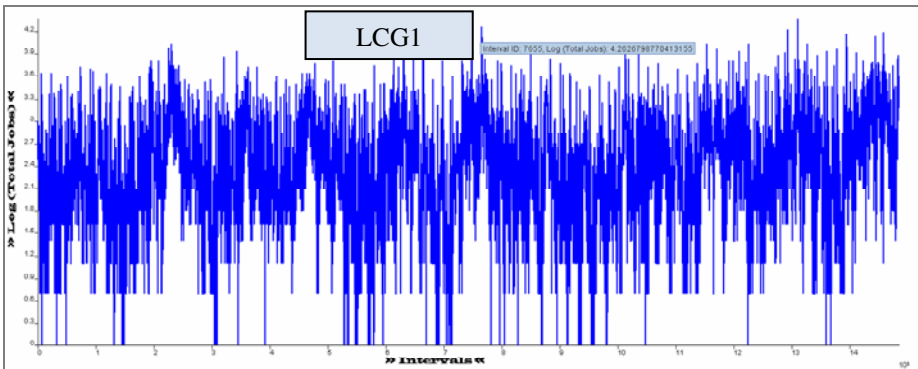


Fig. 10. Job counts for LCG1 and DAS-2

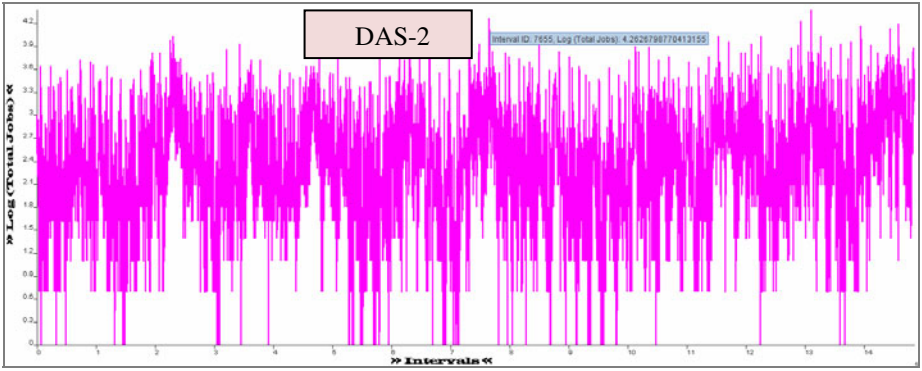


Fig. 10. (continued)

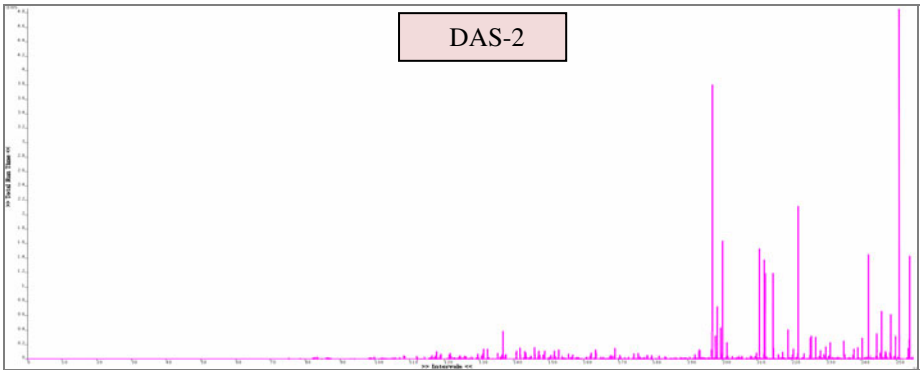
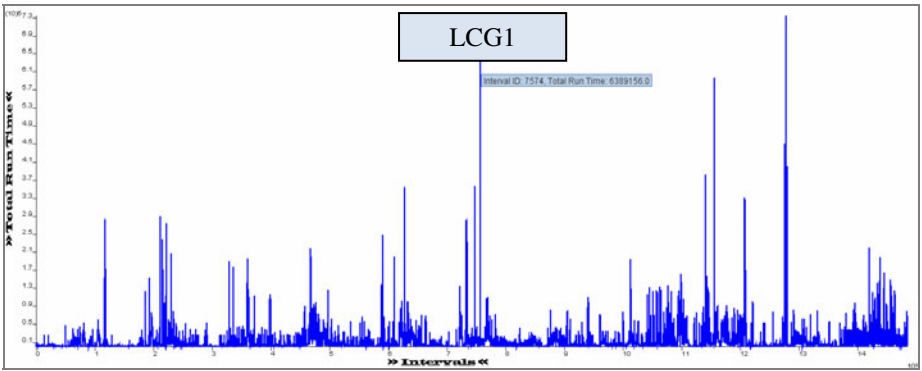


Fig. 11. Total runtime per period, for LCG1 and DAS-2

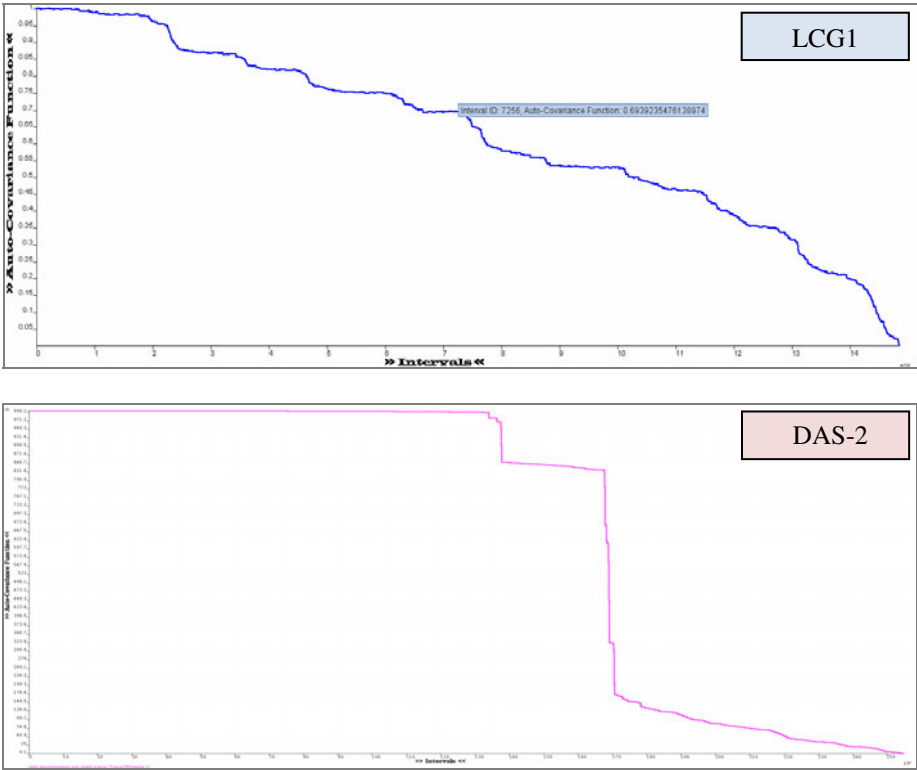


Fig. 12. The autocorrelation function(ACF) of the job counts - LCG1 and DAS-2

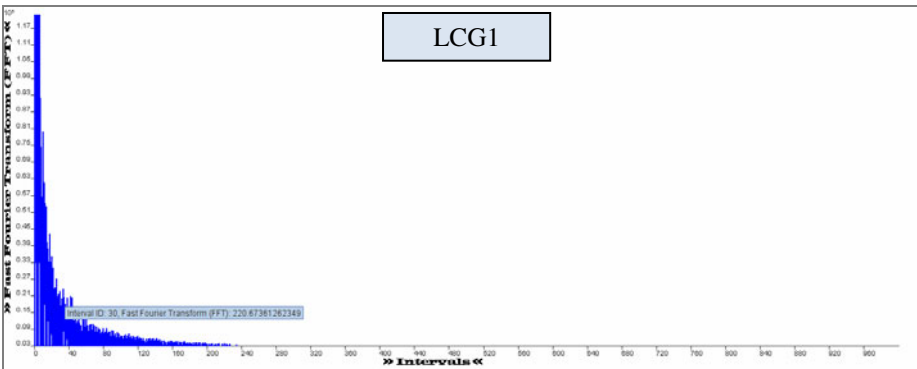


Fig. 13. Fast Fourier transformation(FFT) applied to the autocorrelation of job counts - LCG1 and DAS-2

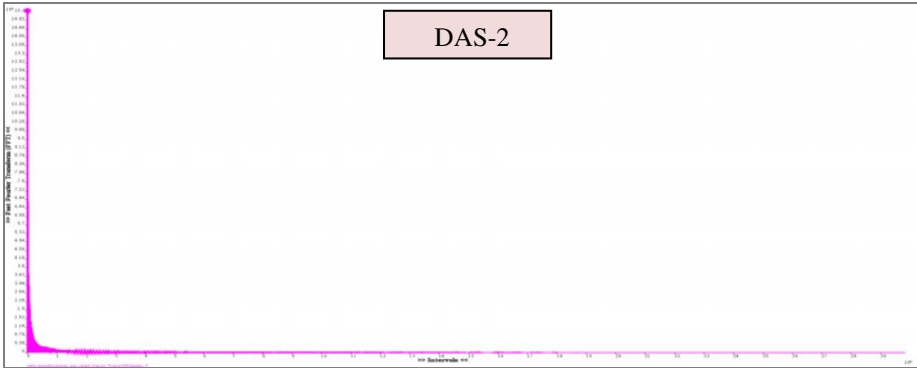


Fig. 13. (continued)

Figure 9 shows the magnitudes of the top 15 groups. Group ‘G1’ in LCG1 had submitted the maximum number of jobs for execution (i.e. 68893).

Figure 10 shows the distribution of job counts for the LCG1 and DAS-2 traces. Figure 11 shows the CPU runtime demand. Next we performed an autocorrelation of the job counts at different lags. Figure 12 shows the autocorrelation plots for LCG1 and DAS-2. We also performed a Fourier analysis by applying the FFT on the values of the autocorrelation output. This is shown in Figure 13. Figures 10 to 13 indicate that job arrivals show a diversity of correlation structures, including short range dependency, pseudo periodicity, and long range dependence. Long range dependence can result in a large performance degradation, whose effects should be taken into consideration for evaluation of scheduling algorithms. The real Grid workloads LCG1 and DAS-2 have shown rich correlation and scaling behavior, which are different from conventional parallel workloads and cannot be captured by simple models such as Poisson or other distribution based methods. LCG1 and DAS-2 will play a key role in the performance evaluation of scheduling algorithms.

7 Conclusion

This paper presents our web based simulator for use in the detailed statistical analysis of real workload traces. The analysis of real workload traces, from both research and production Grids, can aid in a wide variety of parallel processing research. For our experiments, we have analyzed two multi-cluster Grid environments, DAS-2 (from a research Grid) and the LCG1 (from a production Grid), using our web based simulator. We performed a thorough experimentation so as to study the nature of real workload traces. Our simulator allows the user to analyze any real workload trace provided that it is in the Grid Workload Format.

Acknowledgements

We are grateful to the Computer Modeling Lab at University Teknologi PETRONAS for providing the HPC facility to perform the experiments. We want to express our gratitude to Dr. M. Fadzil Hassan, Dr. M. Nordin B Zakaria, Dani Adhipta and Thayalan Sandran from Universiti Teknologi PETRONAS and Dr Aamir Shafi from Massachusetts Institute of Technology (MIT) for their help during the research. We thank the HEP e-Science Group at Imperial College London who provided the LCG data. We also thank Hui Li, the Parallel Workload Archive and the Grid Workloads Archive for their contribution in making the data publicly available.

References

1. Foster, I., Kesselman, C.: *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, San Francisco (2003)
2. Foster, I., Kesselman, C., Tuecke, S.: The anatomy of the grid: Enabling scalable virtual organizations. *International J. Supercomputer Applications*. 15(3) (2001)
3. Yu, D., Robertazzi, T.G.: Divisible load scheduling for grid computing. In: *Proc. Int. Conf. on Parallel and Distributed Computing Systems* (2003)
4. Grid Scheduling Use Cases, <http://www.ogf.org/documents/GFD.64.pdf>
5. Chapin, S.J., Cirne, W., Feitelson, D.G., Jones, J.P., Leutenegger, S.T., Schwiegelshohn, U., Smith, W., Talby, D.: Benchmarks and standards for the evaluation of parallel job schedulers. In: Feitelson, D.G., Rudolph, L. (eds.) *JSSPP 1999, IPPS-WS 1999, and SPDP-WS 1999*. LNCS, vol. 1659, pp. 67–89. Springer, Heidelberg (1999)
6. Ernemann, C., Song, B., Yahyapour, R.: Scaling of workload traces. In: Feitelson, D.G., Rudolph, L., Schwiegelshohn, U. (eds.) *JSSPP 2003*. LNCS, vol. 2862, pp. 166–182. Springer, Heidelberg (2003)
7. Feitelson, D.G.: Metric and workload effects on computer systems evaluation. *IEEE Computer* 36(9), 18–25 (2003)
8. Frachtenberg, E., Feitelson, D.G.: Pitfalls in parallel job scheduling evaluation. In: Feitelson, D.G., Frachtenberg, E., Rudolph, L., Schwiegelshohn, U. (eds.) *JSSPP 2005*. LNCS, vol. 3834, pp. 257–282. Springer, Heidelberg (2005)
9. Parallel Workload Archive, <http://www.cs.huji.ac.il/labs/parallel/workload/>
10. The distributed ASCI supercomputer 2 (DAS-2), <http://www.cs.vu.nl/das2/>
11. Li, H.: Workload dynamics on clusters and grids, *The Journal of Supercomputing* 47(1), 1–20 (2009)
12. Buyya, R.: *Economic-based Distributed Resource Management and Scheduling for Grid Computing*, Ph.D. Thesis, Monash University, Melbourne, Australia (2002)
13. Grid Scheduling Dictionary Project, <http://www.mcs.anl.gov/~schopf/ggf-sched/GGF5/sched-Dict.1.pdf>
14. Baca, D.F.: Allocating modules to processors in a distributed system. *IEEE Transaction on Software Engineering* 15(11), 1427–1436 (1989)
15. Li, H., Buyya, R.: Model-Driven Simulation of Grid Scheduling Strategies. In: *Third IEEE International Conference on e-Science and Grid Computing* (2008)

16. Feitelson, D.G., Rudolph, L.: Metrics and Benchmarking for Parallel Job Scheduling. In: Feitelson, D.G., Rudolph, L. (eds.) IPPS-WS 1998, SPDP-WS 1998, and JSSPP 1998. LNCS, vol. 1459, pp. 1–24. Springer, Heidelberg (1998)
17. Calzarossa, M., Serazzi, G.: Workload characterization: A survey. *Proc. IEEE* 81(8), 1136–1150 (1993)
18. Feitelson, D.G.: Workload modeling for performance evaluation. In: Calzarossa, M.C., Tucci, S. (eds.) *Performance 2002*. LNCS, vol. 2459, pp. 114–141. Springer, Heidelberg (2002)
19. Chiang, S.-H., Vernon, M.K.: Characteristics of a large shared memory production workload. In: Feitelson, D.G., Rudolph, L. (eds.) *JSSPP 2001*. LNCS, vol. 2221, pp. 159–187. Springer, Heidelberg (2001)
20. Feitelson, D., Nitzberg, B.: Job characteristics of a production parallel scientific workload on the NASA ames iPSC/860. In: Feitelson, D.G., Rudolph, L. (eds.) *IPPS-WS 1995 and JSSPP 1995*. LNCS, vol. 949, pp. 337–360. Springer, Heidelberg (1995)
21. Windisch, K., Lo, V., Moore, R., Feitelson, D., Nitzberg, B.: A comparison of workload traces from two production parallel machines. In: 6th Symp. *Frontiers Massively Parallel Computing*, pp. 319–326 (1996)
22. Lublin, U., Feitelson, D.G.: The workload on parallel supercomputers: modeling the characteristics of rigid jobs. *Journal of Parallel and Distributed Computing* 63(11), 1105–1122 (2003)
23. Jann, J., Pattnaik, P., Franke, H., Wang, F., Skovira, J.: J. Riodan.: Modeling of workload in MPPs. In: Feitelson, D.G., Rudolph, L. (eds.) *Job Scheduling Strategies for Parallel Processing*, pp. 95–116. Springer, Heidelberg (1997)
24. Cirne, W., Berman, F.: A comprehensive model of the supercomputer workload. In: *IEEE 4th Annual Workshop on Workload Characterization* (2001)
25. JTransforms,
<http://sites.google.com/site/piotrwendykier/software/jtransforms>
26. JChart2D, <http://jchart2d.sourceforge.net/index.shtml>
27. Commons-Math: The Apache Commons Mathematics Library,
<http://commons.apache.org/math/index.html>

F-IDS: A Technique for Simplifying Evidence Collection in Network Forensics

Eviyanti Saari and Aman Jantan

School of Computer Science, Universiti Sains Malaysia
11800 Penang, Malaysia
eviyanti@fskik.upsi.edu.my, aman@cs.usm.my

Abstract. The increasing numbers of cybercrimes nowadays make network forensic a very important area to be studied. In network forensic analysis, evidence is the crucial elements in the investigation process. However, gathering evidences from network is quite difficult because of the large amount of data in the network system. In addition, getting filtered data for analysis purpose is still a major issue for forensic professional. To contribute in solving the problems, we propose Forensic-based Intrusion Detection System (F-IDS), a new framework to simplify evidences gathering from network by utilizing mechanisms available on the structure of general IDS, the IDS structure will be examined and then enhanced so that the network packet collected by the IDS will be channeled and stored for forensic analysis purpose, also a proper mechanism to identify prospective evidences from the traffic will be proposed. From the conducted system simulation and several testing, the system is able to recognize the expected evidences which are injected as test input based on the classification mechanism.

Keywords: Network forensic; Digital evidence collection; Intrusion Detection System.

1 Introduction

Digital Forensic techniques have gradually changed from traditional methods. According to Wang, J.H, in the past, forensic professionals use fingerprints, DNA typing, and ballistic analysis to make their cases [1]. Nowadays, digital forensic evidences are gathered from medium such as e-mails, internet browser histories, etc.

In general, digital evidence refers to any source or facts that relate the crime with its victim. The United States Department of Justice in the Federal Bureau of Investigation justified that digital evidence can function as target of crime, instrument of crime or repository of evidence that documents the crime itself [2]. Based on the justification, therefore, it is tough to find the most useful digital evidence because of the three roles that digital evidence might play which need to be organized appropriately during investigation.

There are many tools to gather evidence available on the internet whether commercial or open source. Moreover, appropriate tools for investigation are vitally needed to assist forensic professionals in their task and to ensure the reliability and

admissibility of the evidence in court. Research by Erin et.al mentioned that evidence collection methodology need to be improved so that the efforts in term of time and cost will be balanced with the requirements as well as demands from legal community [3]. The main reason for this is the need for using valid and authentic evidences abide by legal authority.

In general, gathering useful evidence from network is tricky and challenging. According to Liao et.al, it is difficult to find useful evidence due to drastically increase in network traffic and large amount of information captured [4]. Consequently, large volume of storage will be used to store the data regardless the importance and accuracy of data as possible evidences. For this reason, we are motivated to help in overcoming this problem by introducing a new framework to help in simplifying evidence collection in the initial stage of evidence gathering.

From our observation, IDS [5-7] could be utilized as a tool for evidence collection. In IDS, sniffer is used as the engine to capture ingress and outgress traffic from network based on packet matching with rule set predefined. We believe if we could manipulate IDS by analyzing the rules in the IDS then we could solve the problem of getting cleaned, filtered and the most accurate evidence in network system. For this reason, this research will propose a new framework for evidence collection in order to obtain useful evidence.

In general, the proposed framework consists of four main elements: evidence collection, evidence identification, evidence classification and the result. Details of this framework will be described in section 3.

2 Background

Network Forensic. Studies in network forensics have become more and more important due to the increasing number of digital crimes along with the popularity of Internet. For example as reported by Malaysia Computer Emergency Response Team (MyCERT), statistics of computer security incident has increased from 922 incidents in Quarter 4, 2009 to 1370 incidents in Quarter 1, 2010 [8]. Specifically, MyCERT reported that incident categorized as System Intrusion has increased dramatically from 404 incidents in Quarter 4, 2009 to 504 incidents in Quarter 1, 2010. According to Pilli et al., the significant increase in security incident and intrusion is the main driving force behind network forensics thus the dire need of forensic professionals to obtain and analyze accurate and useful evidences so that they will admissible in court [9].

In general, a lot of researches have been done on network forensics procedure to produce authentic, accurate, complete and admissible evidence captured from the network system. According to the findings of a survey conducted by Pilli et al., many frameworks proposed by researches implemented different procedures from various models [9]. However, as stated by Trček et al., no firm standards of procedure exist and according to them, this research area is emerging [10]. Therefore, research on a solid standard of forensic procedure is needed to ensure the criteria of the evidence captured comply with the criteria required by legal authority.

Generally, network forensic procedure consists of four main phases as shown in Fig.1: evidence collection, examination, analysis and reporting. These phases are the standard network forensic processes' requirement as defined by National Institute of

Standards and Technology (NIST) [11]. In this paper we will focus on evidence collection phase, which is the initial phase and one of the crucial processes in the network forensic model. Besides, this phase is critical for evidence acceptance in court. Methods on digital evidence collection will be explained in the next section.



Fig. 1. NIST Network forensic processes [11]

Digital Evidence Collection. Evidence collection from the network system or Internet should be the main issue to be focused on as most of the digital crimes happened in these platforms. In order to deal with the evidences collected from Internet, we have to examine the packet from incoming and outgoing traffic. Network packet can be captured by using network monitoring tools like firewall, IDS as well as other network devices.

Firewall can be considered as not so useful to collect evidence due to inability to detect intruders or attack from the network. In fact, according to Rehman, it can only restrict user by using specific rules to deal with incoming and outgoing traffic [12]. Firewall will log all activities or events without the capability to detect if there is intrusion or not. Particularly, if we are to gather evidence from the network by using firewall, we have to check the log files and find out ourselves the suspected packet for useful evidence to be analyzed. However, the log file is inarguably important for forensic professionals to assist them in their forensic task.

There are other network devices which can also be used to collect evidence over the network. Nikkel, for example, presented a small portable network forensic evidence collection device that is built using inexpensive embedded hardware and open source software like OpenBSD 3.8 as depicted in Fig. 2 [13]. However, since the device only operates by request, important data before and after request is not captured and the amount of data collected is limited. Consequently, forensic professional will fail to present the real important evidence that can support their case in court.



Fig. 2. The portable network forensic evidence collector [13]

Intrusion Detection System. In contrast with firewall, IDS is smarter because it has the ability to detect intruders and taking appropriate action against intruders rather than just monitor the network system. IDS will send alert to the network administrator if there is an attack detected in the network. Any attacks, events or activities like port sniffing and packet scanning occur in the network will be stored in the log files. Also, meaningful evidence can be gathered through IDS log files. For this reason, we are motivated to study the use of IDS in assisting task for collecting evidence by focusing on evidence classification for packets which are gathered by the IDS. The next section will describe details on IDS.

In general, there are a lot of IDS available on the internet and one of the popular IDS is Snort. According to Y.H.Choi et al., Snort is the most powerful intrusion prevention and detection system (IDS/IPS) tool in the market [7]. It is a combination of signature and anomaly based techniques. However, the evidence collected by the Snort system is not accurate since the system faced a high rate of false positive and false negative. These problems caused by the limitation of signature based technique that unable to detect new malicious code or virus as supported by Garuba et al. [5]. Therefore, the real important evidence might be overlooked and non-related evidence captured, hence, invalid evidence will be presented in court.

Another open source intrusion detection system is OSSEC. OSSEC is an Open Source Host-based Intrusion Detection System and it performs log analysis, integrity checking, Windows registry monitoring, root kit detection, real-time alerting and active response [14]. According to Daniel, OSSEC can also be categorized in Log-based Intrusion Detection System (LIDS) as it detects attacks by analyzing logs from various sources for example firewall and router [15]. However the integrity of log files has become the issue if the log files get edited by intruders if they successfully infringe the system.

Since packets come continuously to network system for every seconds, minutes, and hours 7 days per week, it is impossible for us to investigate every packet to find the most probable evidence by using IDS, firewall, sniffer, and other tools. The large amounts of network packet come and go from the network system force us to devise a proper strategy to filter or classify the packets as possible evidence or not. For this reason, we will need an appropriate rule set to classify them as well as to increase the accuracy of the collected evidence. Various tools and techniques for collecting evidence will be described in the following section.

Evidence collection tools and techniques. In general, various evidence collection tools and techniques have been developed and implemented. A research by Wang et al. has introduced a dynamical network forensics model based on artificial immune theory and multi-agent theory [16]. This technique is to collect real-time evidence automatically and to provide quick response to network criminals. Moreover, the technique is to overcome infeasibility in analyzing evidence due to manually execution of forensic tool instead of executes the tool automatically. However, this technique intended to collect evidence for further analysis rather than classify the evidence according to the level of evidence possibility. For this reason, evidence classification technique is needed to produce high possibility of evidence.

Other researcher use integration technique in which multiple different approaches are used to generate possible evidence. For example, research by J.Keppens et al. has integrated symbolic crime scenario abduction and Bayesian forensic evidence evaluation to build decision support systems (DSS) for crime investigation [17]. This integration is useful for differentiating competing crime scenarios by which the resulting DSS is capable to formulate effective evidence collection strategies. However, the weakness of this approach is unable to determine time and space of the events in the built scenario since it is important for forensic agent to prepare analysis based on time and space as well.

There also research done on collecting evidence by adopting forensic profiling system. Research conducted by D.Yim et al. for example, proposed the evidence collection of Denial of Service (DoS) attack in wireless environment by applying WLAN Forensic Profiling System [18]. However the rate of packet collected that contain DoS attack by using this approach is lower compared to general IDS. There is the need to improve the efficiency of collecting evidence so that the rate of packet collected will be higher than general IDS.

From the discussion, we observe several problems faced by current tools and techniques: inefficient [18], inability to classify evidence based on level of accuracy [1][5][7] and delay [19]. Arguably, there is the need for better and effective technique that can facilitate evidence collection process in the network to produce accurate evidence. Since IDS is a tool, we believe by manipulating the IDS, we can solve problems discussed previously. Moreover, general mechanism of IDS can be utilized in order to provide the best technique. The next section will introduce the F-IDS framework.

3 Proposed Framework

In general, the proposed F-IDS architecture as shown in Fig. 3 implement distributed mobile agent with centralized manager and also apply redundancy for agents manager to ensure the security and reliability of information collected by agents considering of network or system failure. The implementation of distributed mobile agent is to divide workload between agents so that the task for each agent will be reduced thus, the delay problem will be solved. The data mining engine is for extraction and clustering purpose whereas the evidence repository is to store classified evidence for further investigation or analysis.

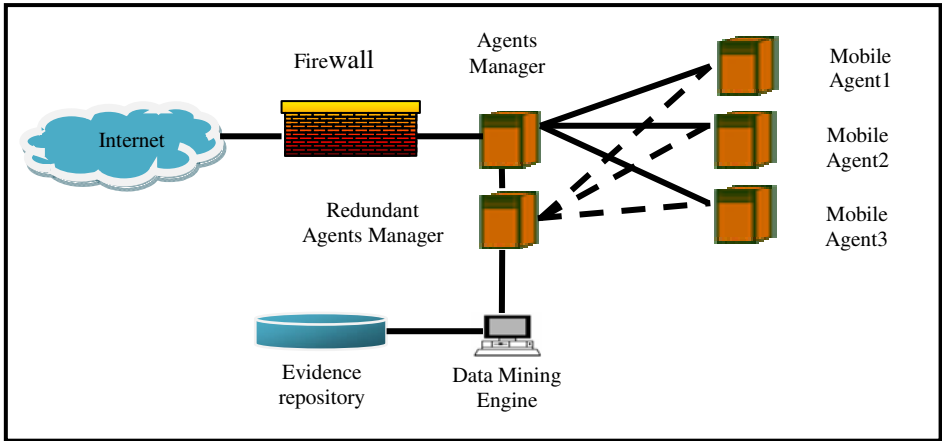


Fig. 3. F-IDS architecture

The framework consists three main phases: evidence collection, evidence identification and classification, result and documentation, as shown in Fig. 4.

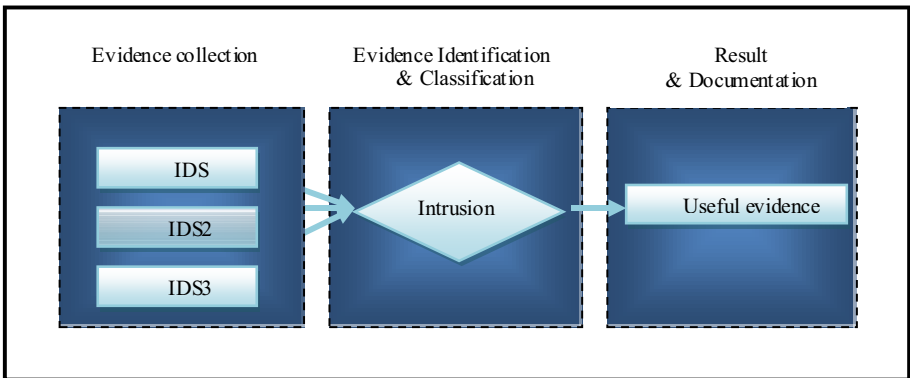


Fig. 4. F-IDS Framework

Phase 1: Evidence Collection

The first phase involves collecting evidence from network system and host by using IDS to collect real time and offline information. The IDS is implemented using hybrid approach by combining anomaly based and signature based IDS with adapting distributed multi agent approach and data mining engine. The rationale of using this approach is as solution for problem faced by previous researchers: overcome problem of false positive and false negative alarm, improve accuracy and detection rate as well as clustering the evidence captured for further analysis. Different IDS rule set is needed by different network users so that the rule set need to be configured

thoroughly to achieve maximum satisfaction result and highly accurate logs. The outcome from this stage is the log file that contains compressed persistence log information from IDS.

Phase 2: Evidence Identification and Classification

At this phase, log analysis will be performed to analyze logs gathered at the first phase. The log analysis is to identify intrusion if the information can be classified as intrusion or not. After that the intrusion will be classified according to level of evidence accuracy so that forensic professionals will have smaller scope of evidence to investigate and analyze. In this phase, all the process will be done by using data mining engine by applying genetic algorithm for clustering purpose. Outcome from this phase is that evidence with high level of possibility and will be stored in the evidence data storage.

Phase 3: Result and documentation

Finally, report of the possible evidence will be generated and documented for further action in court. This process will take into account the custody form that is the basic form for evidence description for analysis.

4 Simulation

Simulation for the proposed model conducted by using system developed in our SRG lab. Fig. 5 depicted the simulation for the testing of F-IDS. There is the need for further experiment and analysis in order to get precise result. The experiment is still ongoing in our lab. We strongly believe that this technique can give big contribution for network forensics in collecting clean and accurate evidence.

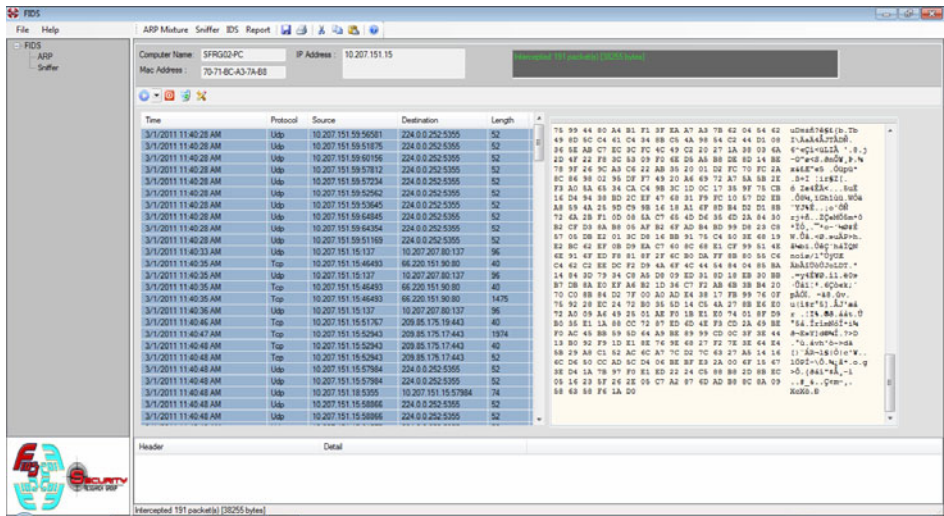


Fig. 5. F-IDS Screen shot

5 Conclusion

The main contribution in this paper is a framework for collecting evidence that can classify the most possible evidence. This framework is also designed to limit the collection of evidence related information so that storage consumption will be reduced. Besides, forensic professional task will be more efficient and effective since the evidence already filtered and classified according to level of possibility. Besides, this research is also being able to contribute in a wise economy by reducing storage space that can reduce hardware costs.

Acknowledgments. This work was supported under “A Forensic Based Detection System to Deal with Digital Evidences from Network” grant no.1001/PKOMP/817048, School of Computer Science, Universiti Sains Malaysia, Penang, Malaysia.

References

1. Wang, J.H.: Cyber Forensics – Issues and Approaches. In: Kumar, et al. (eds.) *Managing Cyber Threats: Issues, Approaches and Challenge*, Kluwer Academic Publishers, Dordrecht (2005)
2. *Digital Evidence Field Guide: What Every Peace Officer Must Know*. Continuing Education Series 1.1. U.S. Department of Justice, Federal Bureau of Investigation
3. Kenneally, E.E., Brown, C.L.T.: Risk sensitive digital evidence collection. *Digital Investigation* 2(2), 101–119 (2005),
<http://www.sciencedirect.com/science/article/B7CW4-4G7X9TV-2/2/9ccac2eb911d64d2570dbcf9837c1d21>,
doi:10.1016/j.diin.2005.02.001
4. Liao, N., Tian, S., Wang, T.: Network forensics based on fuzzy logic and expert system. *Comput. Commun.* 32(17), 1881–1892 (2009),
<http://dx.doi.org/10.1016/j.comcom.2009.07.013>,
doi:10.1016/j.comcom.2009.07.013
5. Garuba, M., Chunmei, L., Fraités, D.: Intrusion Techniques: Comparative Study of Network Intrusion Detection Systems. In: *Fifth International Conference on Information Technology New Generations, ITNG 2008*, April 7-9, pp. 592–598 (2008)
6. Pfleeger, C.F., Pfleeger, S.L.: *Security in computing*, 3rd edn. Pearson Education, Upper Saddle River (2003)
7. Choi, Y.H., Park, J.H., Kim, S.K., Seo, S.W.: An Efficient Forensic Evidence Collection Scheme of Host Infringement at the Occurrence Time. In: *International Conference on Information Security and Cryptology* (December 2006)
8. *CyberSecurity Malaysia. e-Security. Vol: 22-(Q1/2010)* (2010),
http://www.cybersecurity.my/data/content_files/12/692.pdf?diff=1272440150
9. Pilli, E.S., Joshi, R.C., Niyogi, R.: Network forensic frameworks: Survey and research challenges. *Digital Investigation* 7(1-2), 14–27 (2010) ISSN 1742-2876,
<http://www.sciencedirect.com/science/article/B7CW4-4YP6SJY-1/2/10c0909fb97d0954de382e22c99fbc29>,
doi:10.1016/j.diin.2010.02.003

10. Trček, D., Abie, H., Skomedal, Å., Starc, I.: Advanced Framework for Digital Forensic Technologies and Procedures. *Journal of Forensic Sciences* 55, 1471–1480 (2010), doi:10.1111/j.1556-4029.2010.01528.x
11. National Institute of Standards and Technology, Guide to Integrating Forensic Techniques into Incident Response: Recommendations of the National Institute of Standards and Technology (August 2006),
<http://csrc.nist.gov/publications/nistpubs/800-86/SP800-86.pdf>
12. Rehman, R.U.: *Intrusion Detection with Snort: Advanced IDS Techniques Using Snort, Apache, MySQL, PHP, and ACID*. Upper Saddle River, Pearson Education, Inc. (2003)
13. Nikkel, B.J.: A portable network forensic evidence collector. *Digital Investigation* 3(3), 127–135 (2006) ISSN 1742-2876,
<http://www.sciencedirect.com/science/article/B7CW4-4M57H93-1/2/aebf0e677e1c11ae97644dc7e2b02e04>, doi:10.1016/j.diin.2006.08.012
14. Daniel, B., Cid, F.A.Q.: Trend Micro, Inc. (2008-2010),
<http://ossec.net/wiki/Faq:WhatIs>
15. Cid, D.B.: Log Analysis using OSSEC,
<http://www.ossec.net/ossec-docs/auscert-2007-dcid.pdf>
16. Wang, D., Li, T., Liu, S., Zhang, J., Liu, C.: Dynamical Network Forensics Based on Immune Agent. In: *Third International Conference on Natural Computation, ICNC 2007, August 24-27, vol. 3*, pp. 651–656 (2007),
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4344592&isnumber=4344458>, doi:10.1109/ICNC.2007.345
17. Keppens, J., Shen, Q., Schafer, B.: Probabilistic abductive computation of evidence collection strategies in crime investigation. In: *Proceedings of the 10th International Conference on Artificial Intelligence and Law*, pp. 215–225 (2005)
18. Yim, D., Lim, J., Yun, S., Lim, S., Yi, O., Lim, J.: The Evidence Collection of DoS Attack in WLAN by Using WLAN Forensic Profiling System. In: *2008 International Conference on Information Science and Security* (2008)
19. Luo, G., Lu, X., Li, J., Zhang, J.: MADIDS: a novel distributed IDS based on mobile agent. *SIGOPS Oper. Syst. Rev.* 37(1), 46–53 (2003),
<http://doi.acm.org/10.1145/881775.881780>, doi:10.1145/881775.881780

Deterministic-Rule Programs on Specialization Systems: Clause-Model Semantics

Kiyoshi Akama¹, Ekawit Nantajeewarawat², and Hidekatsu Koike³

¹ Information Initiative Center, Hokkaido University, Hokkaido, Japan
akama@iic.hokudai.ac.jp

² Computer Science Program, Sirindhorn International Institute of Technology
Thammasat University, Pathumthani, Thailand
ekawit@siit.tu.ac.th

³ Faculty of Social Information, Sapporo Gakuin University, Hokkaido, Japan
koike@sgu.ac.jp

Abstract. Deterministic-rule programs provide a bridge between rule-based nondeterministic programs and deterministic programs in imperative languages. In this paper, we formulate deterministic-rule programs on a specialization system and formalize their procedural semantics, called the clause-model semantics, based on a general framework for discussing the semantics of procedures with a conditional state-transition structure. The proposed theory establishes a general class of deterministic-rule programs with their precise semantics, providing a basis for developing methods for synthesis of deterministic programs from declarative descriptions. Taking a specialization system as a parameter, the theory is applicable to many concrete classes of deterministic-rule programs with various forms of data structures through parameter instantiation.

1 Introduction

In declarative computation paradigms such as logic programming [8] and functional programming [7], specifications are associated with certain predetermined procedural semantics, allowing one to regard specifications as programs. While they appear attractive, the range of programs possibly generated from a specification in these paradigms is totally restricted, i.e., an obtained program is determined solely by the procedural semantics assigned to the specification. The “specification = program” concept imposes serious limitations on practical program synthesis since it gives the implication that improvement of programs can be achieved mainly by improvement of specifications. As a result, most program transformation methods in the declarative paradigms, including unfold/fold transformation, operate in the specification level, although the term “program transformation” is used. Due to such fundamental reasons, it is difficult to develop in these paradigms a good theory that captures the whole process of synthesizing imperative programs from declarative specifications.

The *equivalent transformation (ET) paradigm* [2,4] was developed with the goal of providing a general basis for a discussion of program synthesis. In this

paradigm, the concept of a program is clearly separated from that of a specification. A specification determines a set of problems of interest and associates declarative meanings with problems. No specific procedural semantics for specifications is assumed. Searching for a program includes not only specification transformation but also a search for a good algorithm, which is realized as a process of incremental construction of a prioritized set of *nondeterministic rewriting rules* (*N-rules*).

A computation state for an N-rule program is represented by a set of definite clauses and a computation is a sequence of states obtained by successive application of N-rules. Computation in the ET paradigm is basically nondeterministic, i.e., a target definite clause in a computation state, target atoms for rule application¹ and an applicable N-rule are selected nondeterministically for making state transition.

To provide a bridge between N-rules and imperative languages, which are normally target languages for program construction, another class of rewriting rules, called *deterministic rewriting rules* (*D-rules*), has been introduced. A D-rule describes one-step transformation of a definite clause by specifying a pattern of a target atom, a rule-applicability condition, and a pattern of replacement atoms, and a *D-rule program* is a sequence of D-rules. D-rules are used in the deterministic computation model of the ET paradigm, where a state is an ordered definite clause, i.e., a definite clause with ordered body atoms, and a target atom for rule application and an applicable D-rule are selected deterministically at each state-transition step based on their appearance order.

For both N-rules and D-rules, the equivalent transformation principle is used as a sufficient condition for computation correctness—if each state transition in a computation is a meaning-preserving transformation step, then the computation is guaranteed to be correct. By its simplicity and generality, this sufficient condition associates many possible correct programs with a specification and greatly widens the possibility of searching for an efficient and correct program.

One important aspect of our previous theory on the semantics of N-rules is its generality [4]. By constructing a theory using only general concepts, one can gain a clear insight into its fundamental and truly important structure, without being distracted by irrelevant details of a specific data domain. The generality of our theory has its roots in the notion of a specialization system [1], which is an abstract axiomatic structure for formulating a space of (extended) atoms and defining a specialization operation on them. In previous works [1,6], a declarative semantics of (generalized) definite clauses, a theory of logical structures, and a procedural semantics of N-rules have all been successfully developed on a specialization system. Taking a specialization system as their parameter, these works are directly applicable to many concrete data domains, e.g., the first-order-term domain, the S-expression domain, and the string domain, by instantiating the specialization system in the respective domain contexts.

¹ Multiple atoms may be selected and together rewritten by an N-rule.

Motivation and Objectives. From the viewpoint of program synthesis, connections between programs and specifications are important and it is essential to design a large class of programs that is suitable for establishment of such connections. Our primary aim in this paper is to formulate a general class of programs for successive transformation of a single definite clause, with a precise procedural semantics. Among many existing programming languages and knowledge representation systems, we take D-rule programs in the domain of S-expressions as the core instances. They form a concrete class of programs for single-clause transformation.

Prolog programs are not programs for single-clause transformation and cannot be used in place of D-rule programs. A pure Prolog program is a sequence of ordered definite clauses, which are regarded as rules for making backward chaining inference. The main structure of the procedural semantics of a Prolog program is a depth-first search for finding all answers to a given query. Such a search corresponds to a process of successively transforming a set of (definite) clauses, rather than single-clause transformation. Moreover, the applicability of a rule (an ordered definite clause) in pure Prolog is determined solely by unification of its head and a target atom; no applicability condition other than unification checking can be specified, resulting in the lack of expressive power for computation control. The extralogical predicate *cut* is introduced in Prolog as a remedy [10]. The sequence of atoms that precedes *cut* in the body of a rule can be considered as the applicability condition of the rule. However, the introduction of *cut* increases difficulty in semantics formalization and theoretical analysis of Prolog programs.

Guarded Horn Clause (GHC) rules [11] yield single-clause transformation and appear to be similar to D-rules. The applicability of a GHC rule is determined by pattern matching and GHC provides commit operators for specifying additional applicability conditions. However, the computation model of GHC is nondeterministic and parallel, while that of D-rule programs is deterministic and sequential.

The semantics of D-rule programs has only been intuitively given thus far. Establishment of a general and rigorous definition of D-rule programs along with their precise formal semantics is a necessary groundwork for discussing deterministic-program synthesis. Our purpose in this paper is to answer the following questions:

1. How to establish a general class of programs for successive transformation of a single definite clause, along with their precise procedural semantics?
2. How to formulate D-rule programs on a specialization system and how to formalize their semantics?

The first problem is set in order to lay the foundations for answering the second problem. To answer the first problem, we establish a general framework for formulating the operational semantics for a class of procedures with a conditional state-transition structure. This semantics defines a computation as a sequence of states in which determination of a state-transition step may require construction

of other computations. To answer the second problem, D-rule programs on a specialization system are then formalized as state-transition procedures, with their semantics being defined by instantiation of the general state-transition semantics framework. By further instantiation of the specialization system, a formalization of D-rule programs in the S-expression domain, which are currently used in our implemented program synthesis system, can be obtained.

Organization and Preliminary Notation. The body of this paper is organized as follows: Section 2 introduces D-rules by means of examples. Section 3 presents our general state-transition semantics framework. Section 4 formulates D-rule programs on a specialization system. Formalization of their procedural semantics is given in Section 5 by applying the semantics framework of Section 3. Section 6 concludes the paper.

The notation below is used. Given a set A , $pou(A)$ denotes the power set of A and $partialMap(A)$ the set of all partial mappings on A (i.e., from A to A). For any sets A and B , $f : A \mapsto B$ denotes a partial mapping f from A to B and $partialMap(A, B)$ the set of all partial mappings from A to B . For any partial mappings f and g , $f \circ g$ denotes the composition of f with g .² For any nonempty sequence s , $first(s)$ denotes the first element of s ; $last(s)$ denotes the last element of s if s is finite; and $rest(s)$ denotes the sequence obtained from s by removing its first element. For any sequences s and s' , if s is finite, $s \cdot s'$ denotes the concatenation of s and s' . Given a set A , $seq(A)$ (respectively, $fseq(A)$) denotes the set of all sequences (respectively, all finite sequences) of elements of A (including the empty sequence).

2 D-Rules by Examples

To begin with, D-rules are introduced by way of examples. For reasons of readability, D-rules in the domain of first-order terms are used.

2.1 Built-In Atoms, Built-In Evaluators, and User-Defined Atoms

In a D-rule system, some atoms, including extralogical atoms, are used for specifying predefined operations, and are referred to as *built-in atoms*. Built-in atoms are evaluated by a predetermined *built-in evaluator*, and if the evaluation succeeds, it yields a substitution as a result. Examples of built-in atoms are equality atoms, i.e., atoms of the form $=(t, t')$, where t and t' are first-order terms. An equality atom represents a unification operation. If t and t' are unifiable, the evaluation of $=(t, t')$ succeeds, with the resulting substitution being their most general unifier.

Atoms other than *built-in atoms* are considered as *user-defined atoms*. User-defined atoms are target atoms for rule application; they are evaluated through definite-clause transformation by applying D-rules. In the examples that follow,

² $f \circ g = \{ \langle x, z \rangle \mid (\langle x, y \rangle \in g) \ \& \ (\langle y, z \rangle \in f) \}$.

$$\begin{aligned}
 r_1: \quad & app([], *Y, *Z) \longrightarrow =(*Y, *Z). \\
 r_2: \quad & app(*a|*X, *Y, *Z) \longrightarrow =(*Z, [*a|*W]), app(*X, *Y, *W).
 \end{aligned}$$

Fig. 1. D-rule examples

Table 1. An example of a transformation sequence using the rules in Fig. 1

Step	Rule	Definite clause
		$ans(*A) \leftarrow app([1, 2], [3], *A)$
1	r_2	$ans(*A) \leftarrow =(*A, [1 *W1]), app([2], [3], *W1)$
2	e	$ans([1 *W1]) \leftarrow app([2], [3], *W1)$
3	r_2	$ans([1 *W1]) \leftarrow =(*W1, [2 *W2]), app([], [3], *W2)$
4	e	$ans([1, 2 *W2]) \leftarrow app([], [3], *W2)$
5	r_1	$ans([1, 2 *W2]) \leftarrow =([3], *W2)$
6	e	$ans([1, 2, 3]) \leftarrow$

atoms of the forms $app(l, l', l'')$, $toSet(l, l')$, $member(t, l)$, and $occur(t, t')$ are user-defined atoms, where l, l', l'' are lists and t, t' are terms. They are intended to mean, respectively, “appending the list l' to the list l yields the list l'' ,” “ l' is the list representing the set obtained from the list l by removing repeated elements,” “the term t is an element of the list l ,” and “the term t occurs in the term t' .”

2.2 Simple D-Rules

Fig. 1 shows two D-rules for rewriting app -atoms, where each variable begins with an asterisk. Their applicability is determined by pattern matching. As specified by its left side, the rule r_1 (respectively, r_2) matches any app -atom whose first argument is the empty list (respectively, a nonempty list). When applied, the rules r_1 and r_2 replace their target app -atoms with corresponding instances of atoms specified in their right sides. Table 1 illustrates transformation of definite clauses by application of these two rules and the built-in evaluator. The predicate ans stands for “answer” and the definite clause in the first row of the table is intended to mean “ $*A$ is the answer if it is the result of appending the list [3] to the list [1,2].” When the first atom in the body of a definite clause is an app -atom, it is rewritten using either r_1 or r_2 . When it is an equality atom, it is evaluated using the built-in evaluator. The label of the rule applied at each rule-application step is shown in the second column of the table. The letter ‘ e ’

$$\begin{aligned}
 r_3: \quad & toSet([], *Z) \longrightarrow =(*Z, []). \\
 r_4: \quad & toSet(*a|*X, *Z), \{member(*a, *X)\} \longrightarrow toSet(*X, *Z). \\
 r_5: \quad & toSet(*a|*X, *Z) \longrightarrow =(*Z, [*a|*W]), toSet(*X, *W). \\
 r_6: \quad & member(*a, [*a|*X]) \longrightarrow. \\
 r_7: \quad & member(*a, [*b|*X]) \longrightarrow member(*a, *X).
 \end{aligned}$$

Fig. 2. D-rule examples

Table 2. An example of a transformation sequence using the rules in Fig. 2

Step	Rule	Definite clause
		$ans(*A) \leftarrow toSet([1, 2, 1], *A)$
1	r_4	$ans(*A) \leftarrow toSet([2, 1], *A)$
2	r_5	$ans(*A) \leftarrow =(*A, [2 *W1]), toSet([1], *W1)$
3	e	$ans([2 *W1]) \leftarrow toSet([1], *W1)$
4	r_5	$ans([2 *W1]) \leftarrow =(*W1, [1 *W2]), toSet([], *W2)$
5	e	$ans([2, 1 *W2]) \leftarrow toSet([], *W2)$
6	r_3	$ans([2, 1 *W2]) \leftarrow =(*W2, [])$
7	e	$ans([2, 1]) \leftarrow []$

Table 3. Checking whether r_4 is applicable in the first step in Table 2

Step	Rule	Definite clause
		$toSet([2, 1], *A) \leftarrow member(1, [2, 1])$
1	r_7	$toSet([2, 1], *A) \leftarrow member(1, [1])$
2	r_6	$toSet([2, 1], *A) \leftarrow []$

in the same column indicates a transformation step resulting from built-in atom evaluation. The transformation changes the initial clause into the unit clause $ans([1, 2, 3]) \leftarrow$, which means “the list $[1, 2, 3]$ is the answer unconditionally.”

2.3 D-Rules with Applicability Conditions

In addition to pattern matching, some applicability condition may be specified in order to confine a rule to being applicable to a more specific class of target atoms, enabling more specific atom replacement. Fig. 2 shows D-rules for rewriting *toSet*-atoms and *member*-atoms. The rule r_4 has an applicability condition, i.e., $\{member(*a, *X)\}$, indicated using a pair of curly braces in its left side. When its target-atom pattern, i.e., $toSet([*a|*X], *Z)$, is instantiated into a body atom using a substitution θ , its corresponding instantiated applicability condition is checked, i.e., $member(*a\theta, *X\theta)$ is evaluated. A separate transformation sequence is constructed for the evaluation. Consider, for example, the definite clause in the first row of Table 2. To check whether r_4 is applicable to the body atom $toSet([1, 2, 1], *A)$, the instantiated condition $member(1, [2, 1])$ is evaluated using the transformation in Table 3. Since this transformation ends with a unit clause, indicating the success of the evaluation, r_4 is applied, resulting in the first transformation step in Table 2.

For each transformation step, a rule is selected deterministically: the applicability of rules are checked one-by-one in the rule appearance order and (only) the first applicable rule is selected. For example, to make the second transformation step in Table 2, the applicability of r_4 to the target atom $toSet([2, 1], *A)$ is checked first, i.e., $member(2, [1])$ is evaluated. Since the evaluation fails, the next rule, i.e., r_5 , which is applicable to the target atom, is used in that step.

$$\begin{aligned}
 r_8: & \text{ occur}(*a, *a) \longrightarrow. \\
 r_9: & \text{ occur}(*a, [*A|*B]), \{\text{occur}(*a, *A)\} \longrightarrow. \\
 r_{10}: & \text{ occur}(*a, [*A|*B]) \longrightarrow \text{occur}(*a, *B).
 \end{aligned}$$

Fig. 3. D-rule examples

Table 4. An example of a transformation sequence using the rules in Fig. 3

Step	Rule	Definite clause
		$ans \leftarrow \text{occur}(1, [[2, 1, 0], [5, 4, 3]])$
1	r_9	$ans \leftarrow []$

2.4 D-Rules with Recursive Applicability Conditions

It is often natural to write a D-rule with a recursive applicability condition. Consider the D-rules for rewriting *occur*-atoms in Fig. 3. The rule r_9 removes an atom of the form $\text{occur}(t_a, [t_A|t_B])$, where t_a, t_A, t_B are terms, if t_a occurs in t_A , i.e., if $\text{occur}(t_a, t_A)$ is satisfied. Table 4 illustrates a transformation sequence using the rules in Fig. 3.³ To determine whether r_9 is applicable to the body atom $\text{occur}(1, [[2, 1, 0], [5, 4, 3]])$, the condition $\text{occur}(1, [2, 1, 0])$ is checked, initiating the transformation sequence in Table 5. To make the first transformation step in Table 5, the applicability of r_9 to $\text{occur}(1, [2, 1, 0])$ is determined, i.e., the condition $\text{occur}(1, 2)$ is checked recursively using the transformation sequence in Table 6. Since none of r_8, r_9 , and r_{10} is applicable to $\text{occur}(1, 2)$, the transformation in Table 6 fails, which is indicated by the symbol ‘ \perp .’ The rule r_9 is thus not applicable in the first step in Table 5 and the next rule, i.e., r_{10} , is used in that step.

2.5 Passing Values through Applicability Conditions

Evaluating an applicability condition may yield a substitution that instantiates body atoms, having the effect of passing a value to them. To illustrate, let us assume that (1) a *calc*-atom is a two-argument built-in atom that takes its first argument as an input for performing some calculation and outputs the calculation result as its second argument, and (2) a *cond*-atom is a one-argument built-in atom that tests whether its argument satisfies a certain condition. Now consider the following two rules:

$$\begin{aligned}
 \text{calcCond}(*x, *z), \{\text{calc}(*x, *y), \text{cond}(*y)\} & \longrightarrow =(*z, *y) \\
 \text{calcCond}(*x, *z) & \longrightarrow =(*z, *x).
 \end{aligned}$$

Suppose that a target atom $\text{calcCond}(t_x, *z)$ is given, where t_x is a term representing an input. To transform this target atom, the applicability of the first rule

³ When it takes no argument, an *ans*-atom means “true.”

Table 5. Checking whether r_9 is applicable in the first step in Table 4

Step	Rule	Definite clause
1	r_{10}	$ans \leftarrow occur(1, [2, 1, 0])$
2	r_9	$ans \leftarrow []$

Table 6. A transformation failure

Step	Rule	Definite clause
1	-	$ans \leftarrow occur(1, 2)$
		\perp

to it is checked first, i.e., a value t_y is produced from t_x using a *calc*-atom and then the condition $cond(t_y)$ is evaluated. If the evaluation of $cond(t_y)$ succeeds, the first rule is applied, i.e., the target atom is replaced with the equality atom $=(*z, t_y)$. If the evaluation of $cond(t_y)$ fails, the second rule is used instead, by which the target atom is replaced with $=(*z, t_x)$.

Supposing that the first rule is applied, the value t_y is in turn passed to $*z$ by the evaluation of the equality atom $=(*z, t_y)$. One may produce the same effect by changing the first rule into

$$calcCond(*x, *z), \{ calc(*x, *y), cond(*y) \} \longrightarrow calc(*x, *z).$$

The application of this new rule does not pass the value t_y to its right side. Instead, it replaces the target atom with $calc(t_x, *z)$, the evaluation of which assigns the value t_y directly to $*z$. This alternative, however, requires two *calc*-atoms to be evaluated—one for rule applicability checking and another for instantiating $*z$ —with their input arguments being the same. If the evaluation of a *calc*-atom takes high cost, the new rule is apparently less efficient.

3 State-Transition Procedures and Their Operational Semantics: A General Framework

Based on the basic structure of the semantics of D-rule programs described in Section 3.1, we establish a general framework for formulating the operational semantics of procedures with a conditional state-transition structure in Sections 3.2-3.4.

3.1 A Basic Structure of D-Rule Program Semantics

A computation of a D-rule program is a state-transition sequence. Checking whether a D-rule is applicable to a given state may involve not only simple

boolean checking of the state's content but also evaluation of user-defined conditions, which requires (possibly recursive) construction of other state-transition sequences based on D-rule application. Such a conditional state-transition structure is an extension of a simple state-transition structure [5], where each step in a state-transition sequence is determined without considering other state-transition sequences. State transition of a D-rule program is characterized by a conditional successor mapping, denoted by $condSucc$. Intuitively, given a set M of computations, $condSucc$ associates with M a partial mapping, i.e., $condSucc(M)$, that determines the successor of a state by considering the effect of the computations in M for evaluating rule applicability conditions. Since a successor state can be known only when all necessary computations required for checking relevant rule applicability are supplied by M , $condSucc(M)$ is in general not total and the mapping $condSucc$ is monotonic with respect to M .

3.2 Computations and State-Transition Procedures

A computation and a deterministic set of computations are defined below:

Definition 1. Let STA be a set of states. A *computation* on STA is a nonempty sequence of states in STA. A computation on STA is said to be *infinite* if it is an infinite sequence, and it is said to be *finite* otherwise. ■

Definition 2. A set M of computations on STA is *deterministic* iff for any $com, com' \in M$, if $first(com) = first(com')$, then $com = com'$. Let $detComp(STA)$ denote the set of all deterministic sets of computations on STA. ■

Next, a general definition of a state-transition procedure is formalized:

Definition 3. A *state-transition procedure* P is an 8-tuple

$$\langle PRB, ANS, STA, INI, FIN, mkSta, mkAns, condSucc \rangle,$$

where PRB, ANS, STA, INI, and FIN are sets, $INI \subseteq STA$, $FIN \subseteq STA$, and

1. $mkSta : PRB \rightarrow INI$,
2. $mkAns : FIN \rightarrow ANS$,
3. $condSucc : detComp(STA) \rightarrow partialMap(STA - FIN, STA)$ such that $condSucc$ is monotonic, i.e., for any $M, M' \in detComp(STA)$, if $M \subseteq M'$, then $condSucc(M) \subseteq condSucc(M')$.

The sets PRB, ANS, and STA are called the *problem space*, the *answer space*, and the *state space*, respectively, of P . Their elements are called *problems*, *answers*, and *states*, respectively. States in INI are called *initial states*, and those in FIN *final states*. The mappings $mkSta$, $mkAns$, and $condSucc$ are called the *making-state* mapping, the *making-answer* mapping, and the *conditional successor* mapping, respectively, of P . ■

In the following, let $P = \langle PRB, ANS, STA, INI, FIN, mkSta, mkAns, condSucc \rangle$ be a state-transition procedure. The mapping $mkSta$ is used for determining an initial state from a given problem, and the mapping $mkAns$ for determining an answer from a final state. Given a set $M \in detComp(STA)$, the partial mapping $condSucc(M)$ associates with a given state its successor with respect to M .

3.3 Operational Semantics

Associated with P is a mapping K_P on $\text{detComp}(\text{STA})$ (Definition 4), based on which the set $\mathcal{M}(P)$ consisting of all state-transition sequences possibly constructed using P can be determined (Definition 5).

Definition 4. For any $M \in \text{detComp}(\text{STA})$, $K_P(M)$ is the set consisting of every computation com on STA that satisfies the following conditions:

1. $\text{first}(com) \in \text{INI}$.
2. For any two successive states st_i and st_{i+1} in com , $\text{condSucc}(M)(st_i) = st_{i+1}$.
3. If com is finite, then $\text{last}(com) \in \text{FIN}$.

For any $n \geq 0$, let $K_P^n(\emptyset)$ be defined by: $K_P^0(\emptyset) = \emptyset$ and if $n \geq 1$, then $K_P^n(\emptyset) = K_P(K_P^{n-1}(\emptyset))$. By Proposition 2, $K_P^n(\emptyset)$ is well defined. ■

Proposition 1. For any $M \in \text{detComp}(\text{STA})$, $K_P(M) \in \text{detComp}(\text{STA})$.

Proof. Let $M \in \text{detComp}(\text{STA})$. For any $st \in \text{STA} - \text{FIN}$, $\text{condSucc}(M)(st)$ is unique when it is defined. $K_P(M)$ is thus deterministic. ■

Proposition 2. For any $n \geq 0$, $K_P^n(\emptyset) \in \text{detComp}(\text{STA})$.

Proof. The result is shown by induction on n . Obviously, $K_P^0(\emptyset)$ is deterministic. Assuming that $K_P^n(\emptyset)$ is deterministic, it follows from Proposition 1 that $K_P^{n+1}(\emptyset)$ is also deterministic. ■

Proposition 3. K_P is monotonic.

Proof. Let $M, M' \in \text{detComp}(\text{STA})$ such that $M \subseteq M'$. Let $com \in K_P(M)$. Since condSucc is monotonic, com also belongs to $K_P(M')$. Thus $K_P(M) \subseteq K_P(M')$. ■

Proposition 4. $[K_P^0(\emptyset), K_P^1(\emptyset), K_P^2(\emptyset), \dots]$ is an increasing sequence.

Proof. Let $n \geq 1$. Since $K_P^n(\emptyset)$ and $K_P^{n+1}(\emptyset)$ are the results of applying K_P n times to \emptyset and $K_P(\emptyset)$, respectively, and $\emptyset \subseteq K_P(\emptyset)$, it follows from Proposition 3 that $K_P^n(\emptyset) \subseteq K_P^{n+1}(\emptyset)$. Since $K_P^0(\emptyset) = \emptyset$, the proposition holds. ■

By Proposition 4, $\lim_{n \rightarrow \infty} K_P^n(\emptyset)$ exists and is equal to $\bigcup_{n=0}^{\infty} K_P^n(\emptyset)$.

Definition 5. The *procedural meaning* of P , denoted by $\mathcal{M}(P)$, is defined as $\lim_{n \rightarrow \infty} K_P^n(\emptyset)$. ■

Proposition 5. $\mathcal{M}(P)$ is deterministic.

Proof. Let $com, com' \in \mathcal{M}(P)$ such that $\text{first}(com) = \text{first}(com')$. By Proposition 4, there exists $n \geq 0$ such that $com, com' \in K_P^n(\emptyset)$. By Proposition 2, $com = com'$. ■

3.4 Computed Answers

Based on finite computations in $\mathcal{M}(P)$, final states are associated with initial states using a partial mapping $reach(P)$. Using $reach(P)$, a partial mapping $exec(P)$ is introduced for determining the answers computed by P .

Definition 6. A partial mapping $reach(P) : \text{INI} \rightarrow \text{FIN}$ is defined as follows: For any $st \in \text{INI}$, if there exists a finite computation com in $\mathcal{M}(P)$ such that $first(com) = st$, then $reach(P)(st) = last(com)$; otherwise $reach(P)(st)$ is not defined. ■

Since $\mathcal{M}(P)$ is deterministic, there is at most one finite computation beginning with st for each $st \in \text{INI}$. Thus $reach(P)$ is well defined.

Definition 7. A partial mapping $exec(P) : \text{PRB} \rightarrow \text{ANS}$ is defined as follows: For any $prb \in \text{PRB}$, if $reach(P)(mkSta(prb))$ is defined, then

$$exec(P)(prb) = mkAns(reach(P)(mkSta(prb)));$$

otherwise $exec(P)(prb)$ is not defined. ■

4 D-Rule Programs on Specialization Systems

After recalling the notion of a specialization system in Section 4.1, we formulate on it a general class of D-rule programs in Section 4.2.

4.1 Specialization Systems

The concept of a specialization system, introduced in [1], provides an axiomatic structure for studying the common interrelations between various forms of extended atoms and specialization operations on them. It provides a basis for a discussion of data structure extension [3] and for formulating declarative descriptions in many data domains, e.g., typed feature terms [9], conceptual graphs [12], and XML expressions [13]. A specialization system is recalled below.

Definition 8. A *specialization system* Γ is a quadruple $\langle \mathcal{A}, \mathcal{G}, \mathcal{S}, \mu \rangle$ of three sets \mathcal{A}, \mathcal{G} and \mathcal{S} , and a mapping μ from \mathcal{S} to *partialMap*(\mathcal{A}) that satisfies the following conditions:

1. $(\forall s', s'' \in \mathcal{S})(\exists s \in \mathcal{S}) : \mu(s) = \mu(s') \circ \mu(s'')$.
2. $(\exists s \in \mathcal{S})(\forall a \in \mathcal{A}) : \mu(s)(a) = a$.
3. $\mathcal{G} \subseteq \mathcal{A}$.

Elements of \mathcal{A}, \mathcal{G} , and \mathcal{S} are called *atoms*, *ground atoms*, and *specializations*, respectively. The mapping μ is called the *specialization operator* of Γ . A specialization $s \in \mathcal{S}$ is said to be *applicable* to $a \in \mathcal{A}$ iff $a \in dom(\mu(s))$. ■

In the rest of this paper, assume that a specialization system $\Gamma = \langle \mathcal{A}, \mathcal{G}, \mathcal{S}, \mu \rangle$ is given. A specialization in \mathcal{S} is often denoted by a Greek letter such as θ . A specialization $\theta \in \mathcal{S}$ is often identified with the partial mapping $\mu(\theta)$ and used as a postfix unary (partial) operator on \mathcal{A} , e.g., $\mu(\theta)(a) = a\theta$, provided that no confusion is caused. Given an expression E containing occurrences of atoms in \mathcal{A} and $\theta \in \mathcal{S}$ such that θ is applicable to all those atoms, we write $E\theta$ to denote the expression obtained from E by replacing each atom $a \in \mathcal{A}$ that occurs in E with $a\theta$. For any $\theta, \sigma \in \mathcal{S}$, let $\theta \circ \sigma$ denote a specialization $\rho \in \mathcal{S}$ such that $\mu(\rho) = \mu(\sigma) \circ \mu(\theta)$, i.e., $a(\theta \circ \sigma) = (a\theta)\sigma$ for any $a \in \mathcal{A}$.

A subset A' of \mathcal{A} is said to be *closed* iff for any $a \in A'$ and any $\theta \in \mathcal{S}$, if θ is applicable to a , then $a\theta \in A'$.

4.2 D-Rule Programs on Specialization Systems

D-rules and D-rule programs on Γ are defined below. Assume that \mathcal{A}_B and \mathcal{A}_U are disjoint closed subsets of \mathcal{A} such that $\mathcal{A}_B \cup \mathcal{A}_U = \mathcal{A}$. Atoms in \mathcal{A}_B are regarded as built-in atoms, and those in \mathcal{A}_U as user-defined atoms.

A *deterministic rule* (for short, *D-rule*) r on Γ is an expression of the form

$$h, \{c_1, \dots, c_m\} \longrightarrow b_1, \dots, b_n,$$

where $h \in \mathcal{A}_U$; $m, n \geq 0$; $c_i \in \mathcal{A}_B \cup \mathcal{A}_U$ for each $i \in \{1, \dots, m\}$; and $b_j \in \mathcal{A}_B \cup \mathcal{A}_U$ for each $j \in \{1, \dots, n\}$. The user-defined atom h is called the *head* of r , denoted by $head(r)$. The sequences $[c_1, \dots, c_m]$ and $[b_1, \dots, b_n]$ are called the *applicability condition* of r , denoted by $cond(r)$, and the *body* of r , denoted by $body(r)$, respectively. The pair of braces on the left side of r does not indicate the set notation; the order of atoms in the applicability condition as well as the order of those in the body of r is important. When $m = 0$, the pair of braces on the left side of r is omitted. A specialization $\theta \in \mathcal{S}$ is said to be *applicable* to r iff θ is applicable to each atom occurring in r . Let $DRULE(\Gamma)$ denote the set of all D-rules on Γ .

A *deterministic-rule program* (for short, *D-rule program*) on Γ is a finite sequence of D-rules on Γ .

5 Clause-Model Semantics

By applying the general state-transition framework developed in Section 3, a procedural semantics of D-rule programs on Γ , called the *clause-model semantics*, is next established. Section 5.1 defines states, initial states, and final states in the clause model. After formalizing a built-in evaluator and rule matching, Section 5.2 identifies D-rule programs with their corresponding state-transition procedures, by means of which their procedural meanings are obtained from the semantics framework of Section 3.

5.1 Clause-Model States

A *clause-model state* (for short, *C-state*) on Γ takes one of the following forms:

1. $ans(as) \leftarrow gs$, where $as, gs \in fseq(\mathcal{A})$.
2. A special symbol \perp , which is called the *null C-state*.

An *initial C-state* on Γ is a C-state on Γ of the first form. A *final C-state* on Γ is either the null C-state (\perp) or a C-state on Γ of the first form such that gs is the empty sequence. Let $STAC$, $INIC$, and $FINC$ be the set of all C-states, the set of all initial C-states, and the set of all final C-states, respectively, on Γ . When no confusion is caused, a C-state is simply called a *state*.

5.2 Clause-Model Semantics

Atom Evaluation and Rule Matching. Assume that \perp is a special symbol not occurring in Γ and it represents an evaluation failure or a pattern-matching failure. A built-in evaluator and rule matching are characterized, respectively, by mappings e_C and m_C , which satisfy the following conditions:

1. $e_C : \mathcal{A}_B \rightarrow (\mathcal{S} \cup \{\perp\})$.
2. $m_C : (DRULE(\Gamma) \times (STAC - FINC)) \rightarrow (\mathcal{S} \cup \{\perp\})$ such that for any rule $r \in DRULE(\Gamma)$ and any state $st = (ans(as) \leftarrow gs) \in STAC - FINC$, if $m_C(r, st) = \theta \in \mathcal{S}$, then θ is applicable to r and $head(r)\theta = first(gs)$.

Conditional Successor Mappings Determined by D-Rule Programs.

Let R be a D-rule program on Γ . A mapping

$$condSucc_C(R, e_C, m_C) : detComp(STAC) \rightarrow partialMap(STAC - FINC, STAC)$$

is defined as follows: Let $M \in detComp(STAC)$ and $st = (ans(as) \leftarrow gs) \in STAC - FINC$. Then:

1. If $first(gs) \in \mathcal{A}_B$, then:
 - (a) If $e_C(first(gs)) = \perp$, then $condSucc_C(R, e_C, m_C)(M)(st) = \perp$.
 - (b) If $e_C(first(gs)) = \sigma \in \mathcal{S}$ and σ is applicable to each atom in as and each atom in gs , then

$$condSucc_C(R, e_C, m_C)(M)(st) = (ans(as\sigma) \leftarrow rest(gs\sigma)).$$

(c) Otherwise $condSucc_C(R, e_C, m_C)(M)(st)$ is not defined.

2. If $first(gs) \in \mathcal{A}_U$, then:
 - (a) For any rule r in R and any $\theta \in \mathcal{S}$ such that θ is applicable to r , let

$$init(r, \theta) = (ans(body(r)\theta) \leftarrow cond(r)\theta).$$

- (b) Determine $condSucc_C(R, e_C, m_C)(M)(st)$ as follows:
 - i. If there exists a rule r in R such that

- $m_C(r, st) \in \mathcal{S}$ and M contains a finite computation com such that $first(com) = init(r, m_C(r, st))$ and

$$last(com) = (ans(bs) \leftarrow []),$$

- and for any rule r' such that r' precedes r in R and $m_C(r', st) \in \mathcal{S}$, M contains a finite computation com' such that $first(com') = init(r', m_C(r', st))$ and $last(com') = \perp$,

then

$$condSucc_C(R, e_C, m_C)(M)(st) = (ans(as) \leftarrow bs \cdot rest(gs)).$$

- ii. Otherwise $condSucc_C(R, e_C, m_C)(M)(st)$ is not defined.

To justify that $condSucc_C(R, e_C, m_C)$ is well defined, first consider the case when Step 2(b)i is used. For any rule r in R , since M is deterministic, if M contains a finite computation com such that $first(com) = init(r, m_C(r, st))$, then com is uniquely determined. Thus, bs is uniquely determined and $condSucc_C(R, e_C, m_C)(M)(st)$ is unique when it is defined at Step 2(b)i. In all other cases, it is obvious that $condSucc_C(R, e_C, m_C)(M)(st)$ is uniquely determined. So $condSucc_C(R, e_C, m_C)(M)$ is a partial mapping.

D-Rule Programs As State-Transition Procedures. A D-rule program R on Γ determines a state-transition procedure

$$\langle PRBC, ANSC, STAC, INIC, FINC, mkStac, mkAnsC, condSucc_C(R, e_C, m_C) \rangle,$$

denoted by $PROC_C(R, e_C, m_C)$, as follows:

1. $PRBC = fseq(\mathcal{A}) \times fseq(\mathcal{A})$.
2. $ANSC = fseq(\mathcal{A}) \cup \{\perp\}$.
3. $mkStac : PRBC \rightarrow INIC$ such that for any $\langle gs, as \rangle \in PRBC$,

$$mkStac(\langle gs, as \rangle) = (ans(as) \leftarrow gs).$$

4. $mkAnsC : FINC \rightarrow ANSC$ such that for any $st \in FINC$,

$$mkAnsC(st) = \begin{cases} as & \text{if } st = (ans(as) \leftarrow []), \\ \perp & \text{if } st = \perp. \end{cases}$$

By identifying the D-rule program R with the state-transition procedure $PROC_C(R, e_C, m_C)$, the procedural meaning of R , denoted by $\mathcal{M}(R)$, is obtained directly from Definition 5, i.e., $\mathcal{M}(R) = \mathcal{M}(PROC_C(R, e_C, m_C))$.

6 Conclusions

In this paper, we develop a general framework for formalizing the meanings of state-transition procedures with a conditional state-transition structure and apply the framework to D-rule programs on specialization systems. The resulting

theory can be instantiated in a specific data domain by (i) instantiating the general concept of specialization system in the context of that domain, (ii) specifying a set of built-in atoms and that of user-defined atoms, and, then, (iii) determining mappings e_C and m_C for evaluation of built-in atoms and for rule matching in the target domain. This work provides well-founded and precise definitions of D-rule programs and their procedural semantics, which are essential for developing a theory for synthesis of D-rule programs from declarative specifications.

Acknowledgment. The work was partly supported by the collaborative research program 2010, Information Initiative Center, Hokkaido University.

References

1. Akama, K.: Declarative Semantics of Logic Programs on Parameterized Representation Systems. *Advances in Software Science and Technology* 5, 45–63 (1993)
2. Akama, K., Shigeta, Y., Miyamoto, E.: Solving Logical Problems by Equivalent Transformation—A Theoretical Foundation. *Journal of the Japanese Society for Artificial Intelligence* 13, 928–935 (1998)
3. Akama, K., Koike, H., Mabuchi, H.: Equivalent Transformation by Safe Extension of Data Structures. In: Bjørner, D., Broy, M., Zamulin, A.V. (eds.) *PSI 2001*. LNCS, vol. 2244, pp. 140–148. Springer, Heidelberg (2001)
4. Akama, K., Nantajeewarawat, E.: Formalization of the Equivalent Transformation Computation Model. *Journal of Advanced Computational Intelligence and Intelligent Informatics* 10, 245–259 (2006)
5. Akama, K., Nantajeewarawat, E.: State-transition Computation Models and Program Correctness Thereon. *Journal of Advanced Computational Intelligence and Intelligent Informatics* 11, 1250–1261 (2007)
6. Akama, K., Nantajeewarawat, E.: Extension of Logical Structures by Safe Extension of Specialisation Systems. *International Journal of Automation and Control* 2, 340–364 (2008)
7. Bird, R.: *Introduction to Functional Programming*, 2nd edn. Prentice-Hall, Englewood Cliffs (1998)
8. Lloyd, J.W.: *Foundations of Logic Programming*, 2nd edn. Springer, Heidelberg (1987)
9. Nantajeewarawat, E., Wuwongse, V.: Defeasible Inheritance Through Specialization. *Computational Intelligence* 17, 62–86 (2001)
10. Sterling, L., Shapiro, E.: *The Art of Prolog*. MIT Press, Cambridge (1986)
11. Ueda, K.: *Guarded Horn Clauses*. Ph.D. Thesis, University of Tokyo (1986)
12. Wuwongse, V., Nantajeewarawat, E.: Declarative Programs with Implicit Implication. *IEEE Transactions on Knowledge and Data Engineering* 14, 836–849 (2002)
13. Wuwongse, V., et al.: A Data Model for XML Databases. *Journal of Intelligent Information Systems* 20, 63–80 (2003)

A Novel Replica Replacement Strategy for Data Grid Environment

Mohammed Madi¹, Yuhanis Yusof¹, Suhaidi Hassan¹, and Omar Almomani²

¹ Universiti Utara Malaysia, Malaysia

s91499@studmail.uum.edu.my, {yuhanis, suhaidi}@uum.edu.my

² Jadara University, Jordan

almomani81@yahoo.com

Abstract. Data Grid is an infrastructure that manages huge amount of data files, and provides intensive computational resources across geographically distributed collaboration. To increase resource availability and to ease resource sharing in such environment, there is a need for replication services. Data replication is one of the methods used to improve the performance of data access in distributed systems. However, it is bounded by two factors: size of available storage and bandwidth of sites within the Data Grid. Hence, there is a need for replica replacement strategy. In this paper, we propose an exponential-based replica replacement strategy (ERRS). OporSim is used to evaluate the performance of this dynamic replication strategy. The simulation results show that ERRS successfully increases data grid performance.

Keywords: Data Grids, Replica Replacement, Exponential Growth/Decay.

1 Introduction

A Data Grid [1] is a geographically-distributed collaboration in which all members require access to the datasets produced within the collaboration. In Data Grids [2, 3], distributed scientific and engineering applications often require access to a large amount of data or they continuously generate several terabytes, even peta-bytes, of raw data in data grid. Therefore one of the tasks in Data Grid is to manage the huge amount of data and facilitate data and resource sharing. In order to achieve this task, data must be copied and stored in several physical locations to vouch the efficient access, without a large consumption of the bandwidth and access latency. In other words, such a system re-quires replica management services that create and manage multiple copies of files. Creating replicas can reroute the client requests to certain replica sites and offer a higher access speed than a single server [4].

Data replication has two direct improvements on the performance of Data Grid. One is to speed up the data access, which leads to a shorter execution time of the grid jobs; and the other one is to save the bandwidth between nodes, which can avoid the network congestion while the sudden frequently requirement of some data. But Replication is also bounded by two factors: the size of storage available at different sites within the Data Grid and the bandwidth between these sites [5]. Sites have

limited storage space and cannot accommodate replicas of every data file on the grid, while network have limited capacity for transferring them. A grid must therefore have a replica management system that manages the data files in grid environment with the aim to optimize the performance of the grid. One of the most important strategies is replica replacement strategy, which is the main focus of this paper. The main rule of replica replacement is to make a room for the newly created replica by finding the victim file (replica) to be replaced by the newly created replica. Replica replacement strategy plays a vital rule in enhancing the performance of grid, therefore, the victim file should be chosen carefully in order to make the right decision. The contribution of the paper is to present a new replica replacement strategy, which is Exponential Replica Replacement Strategy, abbreviated to ERRS. The ERRS improves the temporal locality property [6] and apply the exponential growth/decay model to determine the victim file. The rest of this paper is organized as follows: Section 2 provides a brief description on existing work in dynamic replication strategies and how they determine the victim file. We include the details of our proposed strategy in Section 3 and the performance evaluation is presented in Section 4. Finally, we summarize some conclusions in Section 5.

2 Related Works

In this section, we introduce some of the studies undertaken involving replica replacement strategies. The most well-known replacement policies used commonly in operating systems are: Least Recently Used (LRU) and Least Frequently Used (LFU) [7], which are used in page replacement to free the storage space for more important data. LRU and LFU are examples for this kind of replication strategy that is deployed in Data Grid [7]. In LRU strategy, the requested site caches the required replica, and if the local storage is full or the current free space insufficient for the required replica, the victim replica should be determined and deleted in order to free the storage. The victim replica in LRU is the replica that has the maximum period of time between the current time and the last time the replica was requested. However, in LFU the victim replica is the replica that has less number of requests and also called less popular replica. In [8-10], they proposed a prediction-based replica replacement algorithm using a two-stage process to evaluate the popularity of a replica. They considered some features such as bandwidth and replica size. The simulation results demonstrated that their algorithm contributed to better grid performance. The work in [11] suggested a replica replacement algorithm based on economic model and opportunity cost, the files have been evaluated using zipf-like distribution prediction model and then weighted using the file transfer cost model. If the needed replica has a higher weight than the replica with the lowest weight in the local storage, that file will be deleted and the new replica will be transferred into the local site.

3 The proposed Model

The proposed replica replacement strategy termed as ERRS is applied when the selected site for placing the newly created replica has insufficient storage capacity to store the underlying replica. Prior to that, it is assumed that information on which file to be replicated and where the replica is to be stored is available. ERRS is a strategy

that selects a victim file from the files that are stored in a target storage in order to make sufficient storage space for the underlying replica. The ERRS performs replica replacement through two main stages:

- File Evaluation stage - in this stage we assign a prediction value to each file according to historical information;
- File Elimination - in this stage we eliminate the files from being a victim.

Details of the two stages are included in the following subsections.

3.1 File Evaluation Stage

This work proposes to apply the exponential growth/decay rate in determining value of a file. This is due to the fact that each file has its own number of access and this value increases by the increase of access rate and vice versa. If the access rate increases, so does the growth/decay rate. We describe an exponential growth/decay model for an access number of files in access history. The process of accessing files in data grid environment follows an exponential model. If we use N_f^t to represent the number of access for file f at time t , and N_f^{t+1} to represent the number of access at time $t + 1$, our exponential growth/decay model would be given by:

$$N_f^{t+1} = N_f^t \times (1 + r) \tag{1}$$

Where: r is the growth or decay rate in number of access of a file in one time interval. Therefore, we can calculate the value of r using the following formula:

$$r = (N_f^{t+1}/N_f^t) - 1 \tag{2}$$

First time interval (t_1)		Second time interval (t_2)	
FileID	NOA	FileID	NOA
A	20	A	15
B	17	B	20
C	15	C	13
D	10	D	5
		E	25

Third time interval (t_3)		Fourth time interval (t_4)	
FileID	NOA	FileID	NOA
A	12	A	10
B	24	B	15
C	20	C	30
E	20	E	18

Fig. 1. An example of four time intervals with different number of access of some data files

Assume t is the number of intervals passed, and N_f^t indicates the number of access for the file f at time interval t , then we get the sequence of access numbers:

$$N_f^0 N_f^1 N_f^2 N_f^3 \dots N_f^{t-1} N_f^t$$

Therefore, there are $t - 1$ time intervals, and each time interval has a growth or decay rate in number of access of a file. So, according to the exponential growth/decay model we can write:

$$r_0 = (N_f^1/N_f^0) - 1, \quad r_1 = (N_f^2/N_f^1) - 1, \quad r_2 = (N_f^3/N_f^2) - 1, \\ r_{t-1} = (N_f^t/N_f^{t-1}) - 1 \tag{3}$$

$$\text{Therefore the average rate for all intervals is } r = \sum_{i=0}^{t-1} r_i / t - 1 \tag{4}$$

Having known the average accessed rate (growth or decay) for a file during the past intervals, we can estimate the number of access for upcoming time interval:

$$\text{File Value(FV)} = N_f^t \times (1 + r)$$

In order to avoid extreme cases where the growth or decay rate is equal to infinity, we are assuming that all files have been accessed for at least once. To understand how file evaluation is performed, let us use the data that depicted in Fig. 1 as an example. In this example there are four time intervals with different number of access (NOA) of some data files.

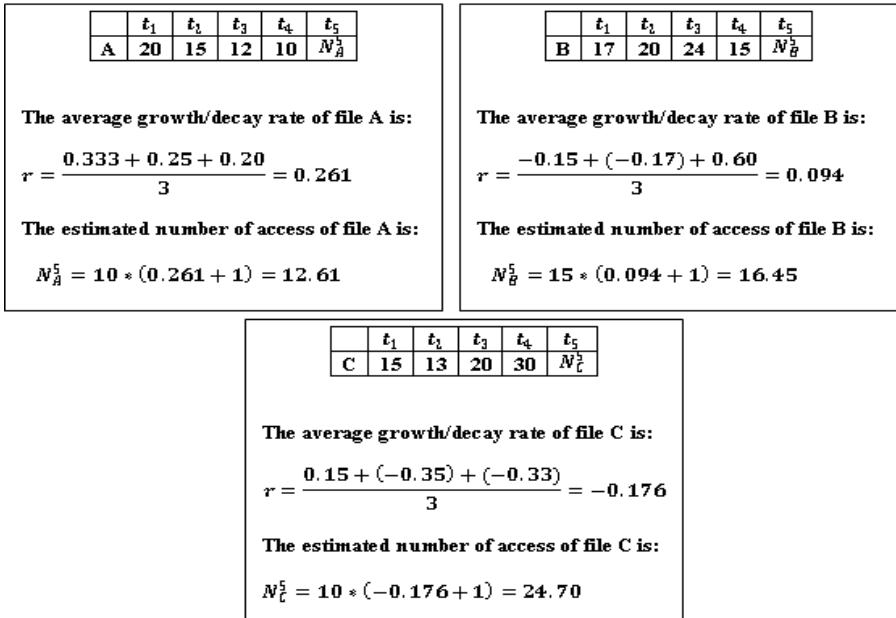


Fig. 2. An example of finding the file value using Exponential model

As shown in Fig. 1, there are five different files accessed during four time intervals. In order to find out the value of each file, we calculate the average growth/decay rate of each file during four time intervals and then substitute in the equation (1). Fig. 2 shows the process of finding the upcoming number of access for files A, B and C. In the same way, we obtained the value of 1.76 as the number of access for file D, and 13.1 for file E. To this end, we determine the estimated number of access of each file, in other words the File Value (FV).

3.2 File Elimination Stage

This stage uses the results of file evaluation stage in order to decide which file to be the victim and which one to be eliminated from deletion. One approach is to select the less valuable file to be the victim for deletion function. However, this approach has a drawback which is the increasing of the number of victim files until there is enough space for the underlying replica, as it depends on the file value. For example, assume that we have 9 files stored in one storage element with free storage space 300 MB as shown in Table 1 , and assume that we have a file with size 1000 MB need to be placed.

In the above example shown in Table 1, the victim file is File2, as it is the less valuable file, but we still need to delete one more file, so the next victim file is File7. So we need to delete two files in order to room one file, in some cases the number of victim files may reach to five as they have small size. That means the system may lose 5 files that are considered stable file to room one file.

Table 1. example of data files stored in on storage element with their corresponding file value and file size.

File name	File Value	File Size
File1	20	400
File2	18	500
File3	23	700
File4	25	1200
File5	24	1100
File6	27	1300
File7	19	900
File8	22	800
File9	28	1500

Another approach is to delete the file, which has a larger size compared to other files. However, this approach is infeasible as in some cases the file that got large size may have the highest value among other files and the system still needs it.

We combine the two approaches together, that means we consider two criteria to determine the victim file: file value (FV) and Storage Cost (SC). The storage cost of a site is the cost of placing a replica in the underlying site. According to [12], storage

cost is the storage space used to store a data file, therefore it depends on the size of the files. The steps of the elimination stage are as follows:

- 1) Sort the files in ascending order based on FV.
- 2) Calculate how much of storage capacity we need in order to room the underlying replica, by applying the following equation:

$$RS = File\ Size - Free\ Space \quad (5)$$

where RS is the required space to host the underlying replica

- 3) Eliminate the files those sizes less than or equal to RS
- 4) Identify the victim file - file that has the lowest FV

To understand how the two approaches combined together, we refer to the example shown in Table 1. We target files that sizes greater than or equal to RS. Therefore, in the above example: $RS = 1000 - 300 = 700$. Completing the elimination stage, that is removing files with sizes less than or equal to 700, we obtain files in Table 2. The file with the least value is File7, hence it will be identified as the victim file.

Table 2. the targeted files for deletion function

File name	File Value	File Size
File7	19	900
File8	22	800
File3	23	700
File5	24	1100
File4	25	1200
File6	27	1300
File9	28	1500

4 Simulation Setup

The evaluation study of ERRS was carried out using OptorSim simulator [13] that adapts the model of EU DataGrid sites and their associated network geometry. In this real-world based workload, a set of high energy physics analysis jobs generated from the Compact Muon Solenoid (CMS) [14] [15] experiments in the European Organization for Nuclear Research (CERN) [16] project. The simulated grid topology used includes 20 sites in USA and Europe as shown in Fig. 2, and the experimental data came from a world wide data production generated in CMS experiments. Within this model, each site, excluding CERN and FNAL, was assigned a Computing and Storage Element. CERN and FNAL were allocated Storage Elements only, as they produce the original files and store them. Jobs were based on the CDF use-case as described in [17], and the other input is simulated the grid jobs and data files configuration that used in [13].

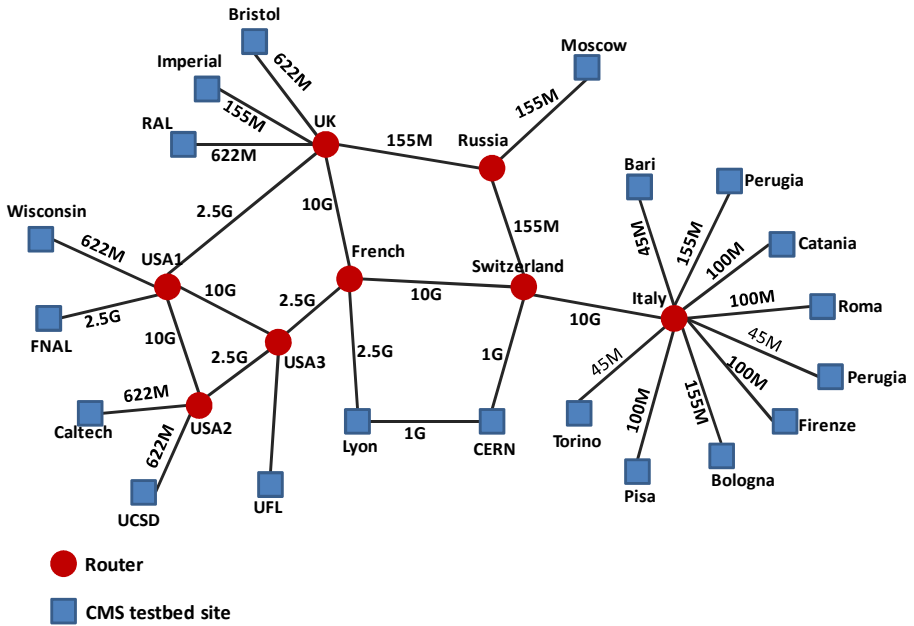


Fig. 2. Grid topology for CMS

The simulation is run for 500 jobs of various types. The jobs are submitted with a fixed probability such that some jobs are more popular than others. Each job is submitted at 25 millisecond intervals. Each job requires specific file for execution. The order of the files accessed in a job is sequential and set in the job configuration file. The number of files in our simulation is 97. The size of the files randomly generated and ranged between 100 MB and 10000 MB.

4.1 Simulation Results

The performance metrics we chose to evaluate the proposed system are: *Mean Job Execution Time (MJET)*, *Efficient Network Usage (ENU)*, and *Hit Rate*. MJET is the average time a job takes to execute, from the moment it is scheduled to Computing Element to the moment when it has finished processing all the required files. ENU is used to estimate the efficiency of network resource usage. It is defined by [13, 18] and calculated by the following equation:

$$ENU = \frac{N_{remote\ file\ access} + N_{replications}}{N_{remote\ file\ access} + N_{local\ file\ access}} \tag{6}$$

Where $N_{remote\ file\ access}$ is the number of accesses that Computing Element reads a file from a remote site, $N_{replications}$ is the total number of file replication occurs, and $(N_{remote\ file\ access} + N_{local\ file\ access})$ is the number of times that Computing Element

reads a file from a remote site or reads a file locally. A lower value indicates that the utilization of network bandwidth is more efficient. The hit rate is the number of times a file request by a job is satisfied by a file which is already present on that site's Storage Element. It indicates the success, or otherwise, of the replication strategies in making as many files as possible available locally, so a higher value indicates that strategy got a higher success. According to [19] the hit rate metric is commonly used for evaluation of the efficiency of replacement strategies. The ERRS is compared with LFU and LRU strategies.

A summary of the results is shown in Table 3. As shown in Fig. 3, the mean job execution time using ERRS is about 15% faster than LFU, and faster about 32% than LRU. The reason of that is because the ESSR invoke the deletion function only one time if there is a need to perform the replacement process. In other words, the process of replacement in ERRS occurs only once as it deletes one file to make a free space for the newly created replica. However, in LRU and LFU the deletion function is invoked many times in one replacement process and need to check every deletion process the storage space of the underlying Storage Element. As a result, LRU and LFU will take longer time to perform the replacement process.

Table 3. Simulation results of ERRS, LFU, and LRU strategies

	ENU (%)	MJET (sec)	Hit Rate (%)
LFU	43.171	57721305	56.801
LRU	39.073	72180999	60.924
ERRS	25.222	48664350	68.550

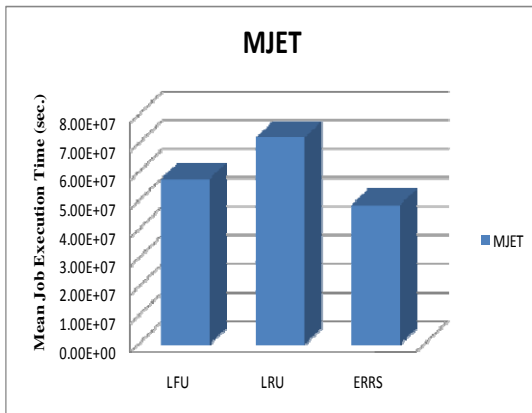


Fig. 3. Mean job execution time

Fig. 4, illustrate the ENU metric, the less ENU the better performance is. Thus the LFU and LRU strategy shows a poor performance in utilizing the bandwidth usage available in the network. However, ERRS can improve the performance about 60% ~ 70%. Moreover, ERRS outperforms LFU by 41% in improving ENU and LRU by 35%. This is because in LRU and LFU number of deleted files resulting from performing the replacement process is large. Therefore, the probability of reading the files remotely will be increased; consequently performing the replication will be increased as well. As a result the ENU will be large. The large number of deleted files by LRU and LFU affects the hit ratio metric as the number of local reads will be decreased as shown in Fig. 5 ERRS outperforms the LFU by 20% and LRU by 12% in the Hit ratio metric.

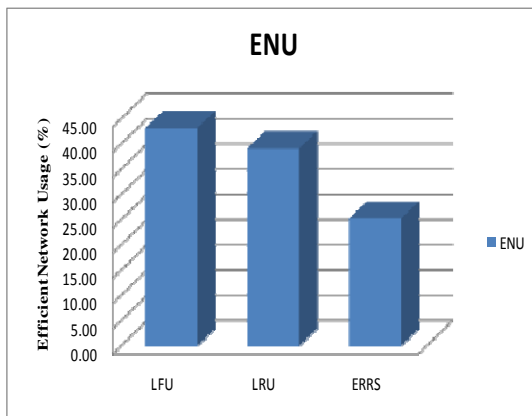


Fig. 4. Effective Network Usage

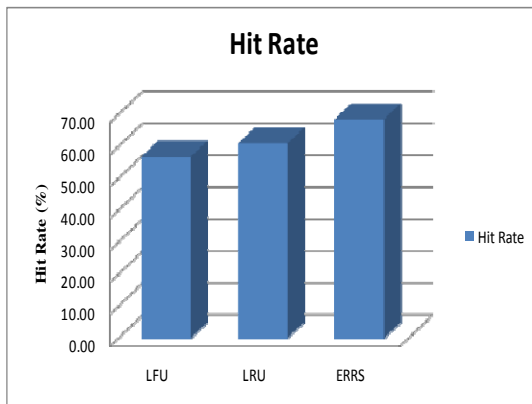


Fig. 5. Hit Rate

5 Conclusion

Exponential growth and decay are mathematical changes. The rate of the change continues to either increase or decrease as time passes. In this paper, we adopted the exponential growth and decay along with storage cost in determining the victim file that need to be replaced. Such an approach considers both the users satisfaction by deleting the less valuable file and resource satisfaction by deleting only one file. Simulation results (via Optorsim) show that the proposed strategy, ERRS, outperformed both LFU and LRU in the measured metrics – mean job execution time, effective network usage and hit rate.

References

- [1] Venugopal, S.: Scheduling Distributed Data-Intensive Applications on Global Grids, PhD thesis, University of Melbourne, Australia (2006)
- [2] Chervenak, A., Foster, I., Kesselman, C., Salisbury, C., Tuecke, S.: The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets. *Journal of Network and Computer Applications* 23 (2001)
- [3] Foster, I.: The Grid: A New Infrastructure for 21st Century Science. *Physics Today* 55, 42–47 (2002)
- [4] Tang, M., Lee, B.S., Yeo, C.K., Tang, X.: Dynamic replication algorithms for the multi-tier Data Grid. *Future Generation Computer Systems* 21, 775–790 (2005)
- [5] Srikumar, V., Rajkumar, B., Kotagiri, R.: A Taxonomy of Data Grids for Distributed Data Sharing, Management, and Processing. *ACM Computing Surveys* 38 (2006)
- [6] Ranganathan, K., Foster, I.: Identifying dynamic replication strategies for a high-performance data grid. In: Lee, C.A. (ed.) *GRID 2001*. LNCS, vol. 2242, pp. 75–86. Springer, Heidelberg (2001)
- [7] Silberschatz, A., Galvin, P.B., Gagne, G.: *Operating System Principles*. Wiley India Pvt. Ltd., Chichester (2006)
- [8] Teng, M., Junzhou, L.: A prediction-based and cost-based replica replacement algorithm research and simulation. In: *19th International Conference on Advanced Information Networking and Applications (AINA 2005)*, pp. 935–940 (2005)
- [9] T. Tian and J. Luo, "A Prediction-based Two-Stage Replica Replacement Algorithm," in *11th International Conference on Computer Supported Cooperative Work in Design, (CSCWD 2007)*, 2007, pp. 594-598.
- [10] Tian, T., Luo, J.: A VO-Based Two-Stage Replica Replacement Algorithm. In: *Network and Parallel Computing*, pp. 41–50 (2010)
- [11] Zhao, W., Xu, X., Xiong, N., Wang, Z.: A Weight-Based Dynamic Replica Replacement Strategy in Data Grids. In: *Asia-Pacific Services Computing Conference*, pp. 1544–1549 (2009)
- [12] Ranganathan, K., Iamnitchi, A., Foster, I.: Improving data availability through dynamic model-driven replication in large peer-to-peer communities. In: *Global and Peer-to-Peer Computing on Large Scale Distributed Systems Workshop*, pp. 376–381 (2002)
- [13] Bell, W.H., Cameron, D.G., Millar, A.P., Capozza, L., Stockinger, K., Zini, F.: Optorsim: A grid simulator for studying dynamic data replication strategies. *International Journal of High Performance Computing Applications* 17 (2003)

- [14] Holtman, K.: CMS data grid system overview and requirements. CMS Note 37 (July 2001)
- [15] CMS Data Challenge (2004), <http://www.uscms.org/s&c/dc04/>
- [16] European Organization for Nuclear Research (CERN),
<http://public.web.cern.ch/Public/Welcome.html>
- [17] Huffman, B.T., McNulty, R., Shears, T., Denis, R.S., Waters, D.: The CDF/D0 UK GridPP Project, CDF Internal Note, vol. 5858 (2002)
- [18] Cameron, D.G., Millar, A.P., Nicholson, C., Carvajal-Schiaffino, R., Stockinger, K., Zini, F.: Analysis of scheduling and replica optimisation strategies for data grids using OporSim. *Journal of Grid Computing* 2, 57–69 (2004)
- [19] Abawajy, J.H.: File replacement algorithm for storage resource managers in data grids. In: *Computational Science-ICCS 2004*, pp. 339–346 (2004)

Investigate Spectrum-Sliced WDM System for FTTH Network

N. Ahmed^{1,*}, S.A. Aljunid¹, R.B. Ahmad¹, Hilal Adnan Fadil¹,
and M.A. Rashid²

¹ School of Computer and Communication Engineering,
University Malaysia Perlis, Malaysia

² School of Electrical Systems Engineering,
University Malaysia Perlis, Perlis, Malaysia
nasim751@yahoo.com

Abstract. In this paper, we have investigated the performance of spectrum-sliced WDM system for FTTH network using NRZ modulation format with 2.5 Gb/s bit rates and 0.4-nm channel spacing. The bit-error-rate (BER) measured against the received optical power is considered as the criterion of the system performance. The simulation results of proposed system were compared with those of conventional system, and found that the NRZ modulation format performs well for 2.5 Gb/s system bit rates. In addition, the total number of multiplexed channels can be increased greatly in spectrum-sliced WDM technique. Therefore, 0.4-nm channel spacing spectrum-sliced WDM system is well suited for the Fiber-to-the-Home (FTTH) network.

Keywords: LED, WDM, spectral slicing, NRZ modulation format, FTTH.

1 Introduction

More than one decade wavelength-division-multiplexed (WDM) system is being used as an attractive technology for optical access network. The number of advantages, including a large capacity, network security and bit rate transparency make WDM system is a first choice for optical access network [1]-[3]. However, this technology requires a several costly laser diodes at the transmitter side with different wavelengths [2]. Because of the costly laser diodes this brings many issues such as wavelength management and installation cost [4]. Therefore, a number of approaches are proposed to reduce the wavelength cost such as spectral slicing. A conventional WDM system can provide only moderate data rates. Therefore, a conventional WDM system is not suitable for the systems that require high data transmission rate regardless of short haul or long haul transmission line. Recently, the demand of transmitting high data rates and narrow channel spacing are increasing dramatically with the increase in using optical local access networks, like the Metro Area Network (MAN), Fiber-to-the-Home (FTTH) or the Fiber-to-the-Building (FTTB) depending on the location of Passive Optical Networks (PON's) terminals [5]-[6].

* Corresponding author.

The primary requirement for these applications is the channel bit rates of several Mb/s over a distance of several kilometers. Therefore, it is becoming a challenging issue to develop a new technology with low cost that can accommodate the current needs of high transmission data rate. Consequently, the researchers and scientists introduced a technology called "Spectral Slicing" that brings the WDM technology one step more ahead of advanced technology [7]-[8]. In this technology, the low cost Light Emitting Diode (LED) is used to play the important role of light sources instead of costly laser diodes. Since LED has very broadband spectrum then this spectrum can be sliced spectrally into many independent signals with equal channel spacing. Spectral slicing is an attractive wavelength division multiplexing (WDM) technique that provides a means of sharing a multiwavelength optical source amongst many users [9]. However, to select the optimum channel spacing between the two sliced spectral is a very critical task. To date, 0.8-nm channel spacing is used in spectral-sliced WDM system experimentally [12]. However, 0.4-nm channel spacing for spectral-sliced WDM has not yet been investigated. Furthermore, wavelength controlled arrays of distributed-feedback (DFB) lasers; spectrally sliced light-emitting diodes (LED's) have been used in experimental WDM systems [3].

In this paper, we have investigated a 4-channel spectral-sliced WDM system using 0.4-nm channel spacing and LED as a broad band source. We have evaluated the effectiveness of narrow band WDM system for FTTH by simulation. We can increase the capacity of the system more by reducing the channel spacing but leads to a severe cross-phase modulation (XPM) problem [10]-[11]. In order to keep this effect under tolerance level, we choose 0.4-nm channel spacing. This method is particularly attractive for the local access and metro area network since it eliminates the need for expensive, wavelength controlled transmitters on the consumer side of the network. We use only one LED as a multi-wavelength source and use the spectrum slicing technique to realize the WDM system for a shot haul transmission. The paper is organized as follows: The system architecture of the proposed system is shown in Sec 2. Sec 3 is devoted for the network simulation. The results and discussions are discussed in Sec 4, and finally conclusions are drawn in Sec 5.

2 System Architecture

Fig.1 shows the proposed architecture of 4-channel spectral-sliced WDM system with 0.4-nm channel spacing. We used a single broad band light source (LED) which carries 1550 nm wavelength. The LED emits between -1.283 dBm to 9.51 dBm input power for the system with bit rates at 2.5 Gb/s.

In the transmitter side of the proposed system, the LED's broad spectrum is sliced into four input signals by a demultiplexer. Then the demuxed signal is modulated using the modulator and the multiplexer combine the entire modulated signal. The sliced signals are modulated by the $2^{15}-1$ pseudorandom bit sequence and then the modulated signal transmitted over 50km standard single mode fiber (SMF). In the receiver side, the demultiplexer combines all the input signals. The signal is detected by photo detectors (PIN). We considered the attenuation loss of the fiber as 0.25 dB/km and the insertion loss as 0.2 dB. Finally, we analyze the BER signal using eye-diagram analyzers to evaluate performance of each channel.

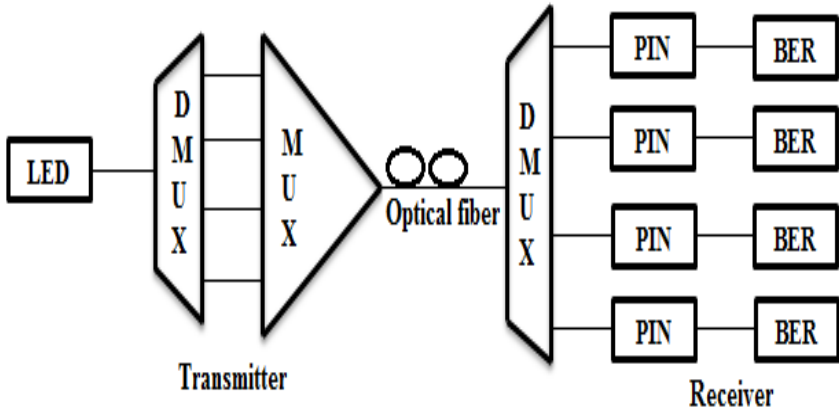


Fig. 1. Schematic diagram of 4-channel spectral-sliced WDM systems with 0.4-nm channel spacing.

3 Network Simulation

We have conducted the simulation accordance with the simulation setup as shown in Fig.2. The simulation software Optisystem, version 9.0 was used. We run the simulation with 2.5 Gb/s bit rates and 0.4-nm channel spacing. The slice is modulated using Mazh-Zehnder modulator and pseudorandom bit sequence. The Non-Return-to-Zero (NRZ) modulation format was chosen for coding due because of its simplicity. The simulation was run for 50 km distance using (SSMF) standard single mode fiber. A single mode

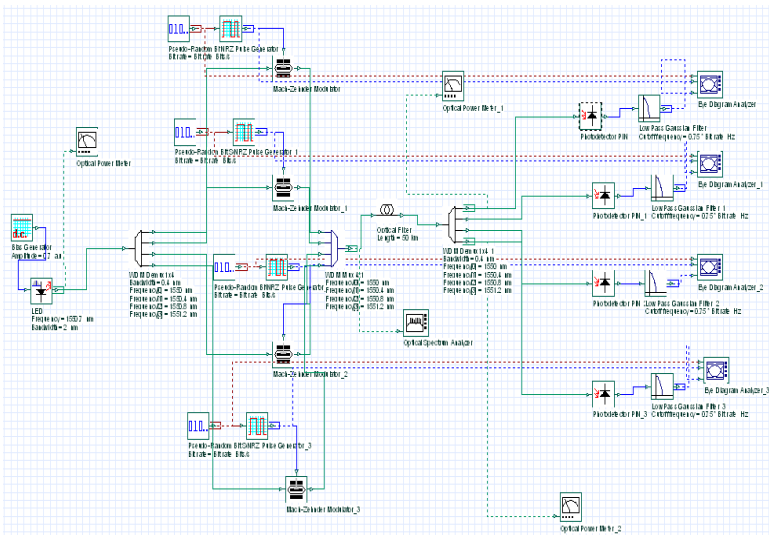


Fig. 2. Simulation setup of the proposed transceiver scheme

fiber was chosen to minimize the dispersion effect. The zero-dispersion of wavelength and the dispersion slope of DSF were 1550 nm and $0.075 \text{ ps}/\sqrt{\text{km}\cdot\text{nm}}$. The average fiber loss was considered about 0.2 dBm including the splicing loss.

On the receiver side of the system PIN photodetector is used to detect the signal with electrical low pass Gaussian filtering. Each channel performance is measured using BER meters. The optical spectrum analyzer and optical power meter is used to detect the channel signal and calculate the input/output power as the received power. The optical signal input power was from -1.283 dBm to 9.51 dBm, whereas the received power was -18.896 dBm to -25.886 dBm.

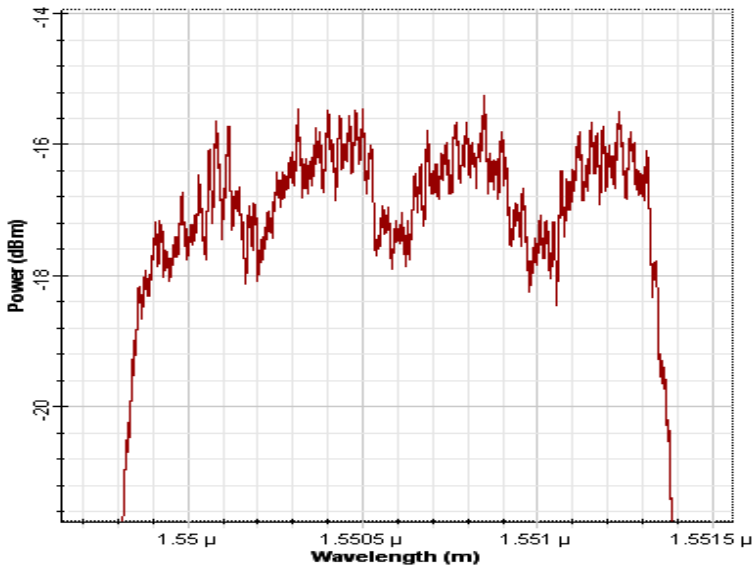


Fig. 3. Spectrum of light source (LED) using demultiplexer for 4 channels at 2.5 Gb/s system

Fig. 3 shows the optical spectrum-sliced signal. The multiplexer and demultiplexer with same configuration were used for slicing a signal. The simulation parameters such as attenuation, group delay, group velocity dispersion, dispersion slope, and non-linear effects are activated during simulation. We run the simulations for the proposed system of 0.4-nm channel spacing and the conventional system of 0.8-nm channel spacing. We summarize the simulation results on both proposed and conventional systems in Table 1 for comparison.

Table 1. Simulation results of 0.8-nm and 0.4-nm channel spacing

Channel Spacing (0.8-nm)		Channel Spacing (0.4-nm)	
Received Power	BER	Received Power	BER
-25.239	3.11e-12	-25.886	5.97e-009

4 Results and Discussion

The system performance has been characterized using the bit-error-rate (BER) against the received optical power and the eye pattern. Fig.3 shows the comparison of results between 0.8-nm and 0.4-nm channel spacing. In order to clarify more clearly, the obtained results are summarized in Table 1. It is seen from Table 1 that the received power of 0.4-nm channel spacing is -25.886 dBm whereas -25.239 dBm for 0.8-nm channel spacing.

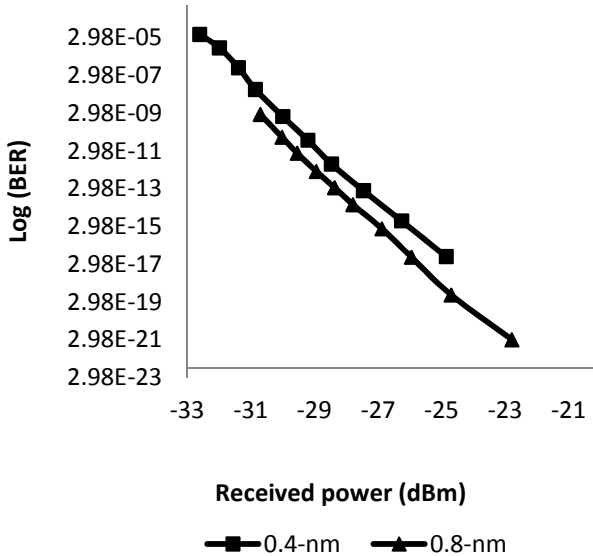
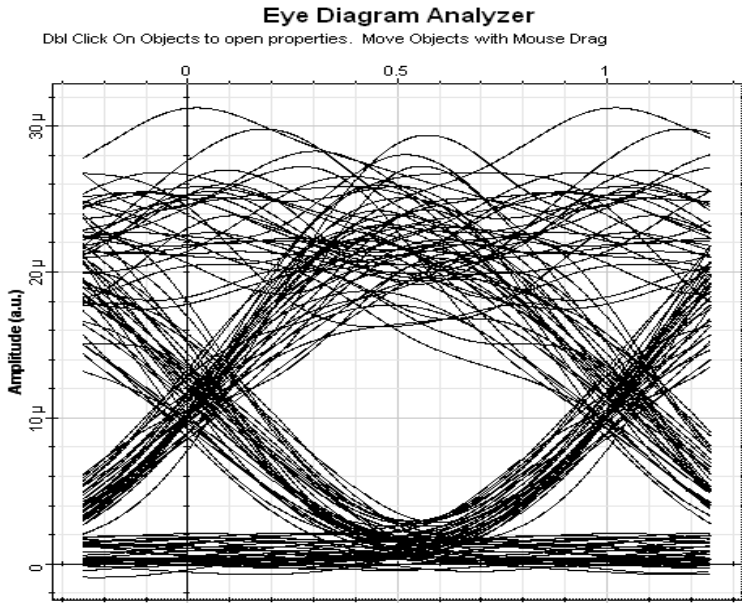


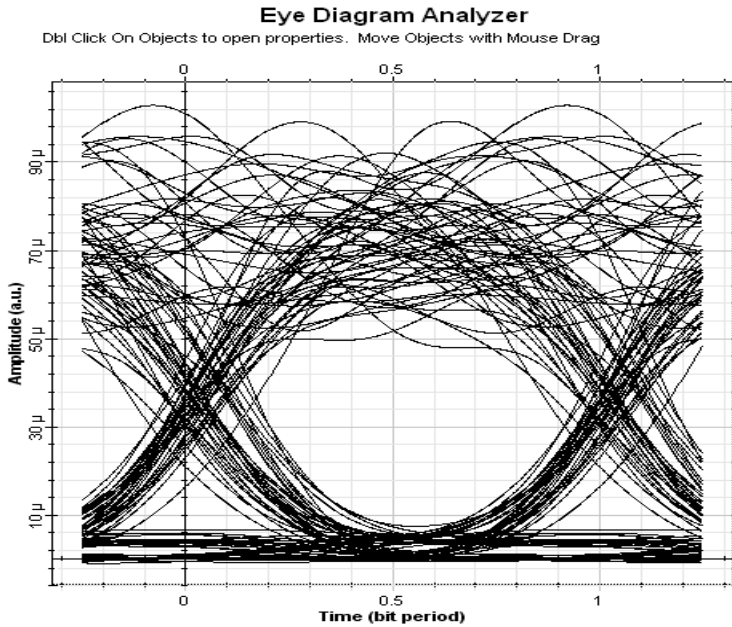
Fig. 3. BER curves measured at ch.1 with all four channels transmitting for 50-km fiber with 0.4-nm and 0.8-nm channel spacing, where NRZ modulation format is used.

It is natural that 0.4-nm channel spacing received more power compared to 0.8-nm due to the less usages of light source bandwidth. When then the received power decreases the BER become worse. On the other hand, the quality of signal with 0.8-nm channel spacing is slightly better compared to 0.4-nm, because the beat noise occurs between two channels due to narrow channel space. However, the BER of small channel spacing 0.4-nm still satisfies the standard acceptable value (less than 10^{-9}).

Furthermore, the eye diagram shown in Fig.4 also illustrate that the proposed system performs well and suitable for FTTH at bit-rates 2.5 Gb/s. The 0.4-nm channel spacing spectrum-sliced WDM system can transmit the signal without any error, and can make the system more stable. We can then conclude that the proposed spectrum-sliced WDM system with 0.4-nm channel spacing is cost effective and well suited for FTTH network.



(a) 0.4-nm channel spacing



(b) 0.8-nm channel spacing

Fig. 4. Received eye diagrams measured at the receiver of NRZ modulation format with 0.4 and 0.8-nm channel spacing

5 Conclusion

We have investigated a low cost WDM system with 50 km single mode fiber at 2.5 Gb/s bit rate using spectral slicing technique. In this investigation, the proposed system is compared with the conventional WDM system. Although the quality of signal for the proposed system (0.4-nm channel spacing) is slightly worse than those of conventional system (0.8-nm channel spacing), the BER of proposed system still satisfy the standard acceptable value (smaller than 10^{-9}). Moreover, the received optical power of the system is better than conventional system. Therefore, the proposed system would be the cost effective and suitable for the optical access networks, such as Fiber-to-the-Home (FTTH).

References

- [1] Yoshida, S., Kuwano, S., Takachio, N., Iwashita, K.: 10 Gb/S C₁₀ Channel WDM Transmission Experiment Over 1200 Km With Repeater Spacing 100 Km Without Gain Equalization Or Pre-Emphasis. OFC. Paper, TuD6 (1996)
- [2] Dutta, A.K., Dutta, N.K., Fujiwara, M.: WDM Technologies, Active Optical Components, p. 315. Academic press of Elsevier Science, Oxford (2004)
- [3] Lee, C.H., Sorin, W., Kim, B.Y.: Fiber to the home using a PON infrastructure. Journal of Lightwave Technology 24(12), 4568–4583 (2006)
- [4] Lee, H.-K., Lee, H.-J., Lee, C.-H.: A low cost WDM-PON using spectrum sliced F-P LDs with Cyclic Arrayed Waveguide Gratings. In: COIN - ACOFT 2007, Melbourne, Australia (June 2007) ISBN 978-0-9775657-3-3
- [5] Thiele, H.J., Killely, R.I., Bayvel, P.: Influence Of Transmission Distance On Xpm-Induced Intensity Distortion In Dispersion Managed, Amplified Fiber Links. Electron. Lett. 35, 408–409 (1999)
- [6] Wagner, S.S., Lemberg, H.L.: Technology And System Issues For A Wdm-Based Fiber Loop Architecture. Journal Of Lightwave Technology, 1759-1768 (1989)
- [7] Reeve, M.H., Hunwicks, A.R., Zhao, W., Methley, S.G., Bickers, L., Hornung, S.: Led Spectral Slicing For Single-Mode Local Loop Applications. Electron. Lett. 24, 389–390 (1988)
- [8] Lee, J.S., Chung, Y.C., Giovanni, D.J.D.: spectrum-Sliced Fiber Amplifier Light Source For Multi-Channel WDM Applications. IEEE Photon. Lett., 1458–1461 (May 1993)
- [9] Zimgibl, M., Joyner, C.H., Stulz, L.W., Dragone, C., Presby, H.M., Kaminow, I.P.: Larnet, A Local Access Route1 Network. IEEE Photon. Lett., 215–217 (July 1995)
- [10] Shtaif, M., Eiselt, M., Tkach, R.W., Stolen, T.H., Gnauck, A.H.: Crosstalk In Wdm Systems Caused By Cross-Phase Modulation In Erbium-Doped Fiber Amplifiers. IEEE Photon. Lett. 12, 1796–1798 (1998)
- [11] Shtaif, M., Eiselt, M., Garrett, L.D.: Cross-Phase Modulation Distortion Measurement In Multispan Wdm Systems. IEEE Photon. Lett., 88–90 (2000)
- [12] Eiste, M., Gorrett, L.D., Tkach, R.W.: System Performance Of An Eight-Channel Wdm Local Access Network Employing A Spectrum-Sliced And Delay-Line-Multiplexed Led Source. IEEE Photonic. Lett. 9, 696–698 (1997)

Use Case-Based Effort Estimation Approaches: A Comparison Criteria

Mohammed Wajahat Kamal^{*}, Moataz A. Ahmed, and Mohamed El-Attar

Information and Computer Science Department,
King Fahd University of Petroleum and Minerals,
Dhaharan 31261, Saudi Arabia
{wajahat, moataz, melattar}@kfupm.edu.sa,
wajju.999@gmail.com

Abstract. Reliable and accurate software development effort estimation has always been a daunting task for project managers. More recently, the use of Use Cases for software effort estimation has gained wide popularity. Researchers from academia as well as industry have shown interest in the Use Case based approaches because of the promising results obtained along with their early applicability. There has been a number of approaches proposed in the literature. However, there is no framework that could be used to aid practitioners in selecting appropriate approaches suitable for their particular development efforts. In this paper we present an attribute-based framework to classify and compare these approaches and provide such aid to practitioners. The framework is also meant to guide researchers interested in proposing new use case-based approaches. The paper discusses a number of representative Use Case-based effort estimation approaches against the framework. Analysis of the discussion highlights some open issues for future research.

Keywords: Effort estimation, Use Case, Comparison Criteria, and UML.

1 Introduction

Effort is delineated as the amount of labor required to complete a certain work. Software effort estimation is the process of predicting the effort required to develop a software system based on incomplete, crude, uncertain or ambiguous inputs [12], [15]. It deals with the prediction of the most probable cost and time to actualize the required development task. Software effort estimation spawned some of the first attempts at rigorous software measurement, so it is the oldest, most mature aspect of software metrics. Researchers have proposed so many models to be used for effort estimation. One of the main inputs to any effort estimation model is the estimated or the actual software size, e.g., lines of code (LOC). As such, measuring/estimating the software size accurately and also as early as possible is of prime importance [7], [16]. A good size estimate can lead to a good effort estimate. This is a challenging task though, since on one hand, early effort estimates play a vital role when bidding for a

^{*} Corresponding author.

contract or determining whether a project is feasible in terms of a cost-benefit analysis [5], [17], [23], [29]. On the other hand, however, early estimates of size, for example based on requirements specification, are the most difficult to obtain, and they are often the least accurate, because very little detail is known about the project and the product at its start [8]. Furthermore, available details are characterized as being imprecise and uncertain.

Use cases, as being available relatively early during the software development lifecycle, are expected to offer a good estimate of the size of the corresponding future system. Consequently, effort estimation using *use cases* has been gaining popularity and the response to the approach has been received quite well by the research community. Some metrics along with corresponding techniques have been proposed to estimate effort using use case information. Majority of them utilize the basic Use Case Points [14] size metric as a predictor of effort. In our bid to carry out a critical survey of the literature on using *use cases* for software development effort prediction, we discovered that a common ground for assessing and comparing these prediction techniques is not available. Though a few related works are available, there is no significant contribution which explicitly offers an evaluation framework for comparison and evaluates the proposed Use Case based metrics on a common platform [3], [10], [25], [28]. Boehm [4] presented a set of useful criteria (attributes) for evaluating the utility of software cost models. The attributes targeted model-based estimation methods. Similarly, Saliu and Ahmed [26] proposed a set of attributes; theirs targeted soft computing-based effort estimation models though. As such, no criteria were developed to target use case-based models. The primary goal of this work is to fill the void caused by the unavailability of such literature which can help practitioners in selecting appropriate metrics for their respective development efforts and also guide researchers interested in developing new metrics in this domain. Accordingly and based on a comprehensive survey, we identified some set of attributes to be used in assessing and comparing various use case-based approaches for effort prediction. This set of attributes is presented in Section 4.

The rest of paper is organized as follows: Section 2 gives a brief insight about the paradigms and problems associated with using Use Cases. A brief summary of the metrics included in the study is presented in Section 3. Section 4 presents the comparison framework and definitions of the attributes. Section 5 consists of the comparison tables and the actual comparison of the available Use case metrics. Section 6 is the analysis of the comparison findings. Section 7 concludes the paper and presents plans for future work.

2 Use Case-Based Effort Estimation

The history of using use cases for effort estimation started with the development of the Unified Modeling Language (UML) by Jim Rumbaugh, Grady Booch, and Ivar Jacobson of Rational Software Corporation in mid-nineties [13]. Sometime later, UML was incorporated into the Rational Unified Process RUP by Rational Software. Meanwhile, Gustav Karner also of Rational Software Corporation developed an estimating technique to predict the effort required based on Use Cases, much the same

way as Function Points. Karner's technique is known as Use Case Point Method [14] and is incorporated into RUP. It is the basic estimating technique for predicting effort based on use cases.

Use cases are used to capture and describe the functional requirements of a software system. Use Case Models define the functional scope of the system. The Use Case model is relevant and valuable for early size measurement and estimating effort as it employs use cases as input. According to a survey conducted by Neil and Laplante [20], 50% of the software projects have their requirements presented as Use Cases. Based on these facts, the approach to estimate effort using Use Cases has gained popularity and subsequently the basic technique proposed by Karner, UCP has gained more recognition. The idea is more or less same as the Function Points developed by Albrecht [2]. Based on UCP, many techniques have been proposed since then, like Use Case Size Points [6], Extended Use Case Points [30], UCP modified [9], Adapted Use Case Points [18], Transactions [24] and Paths [24] to mention a few. A more detailed description of the aforementioned techniques will be presented in the later sections.

Along with the advantages of using these methods, several issues and concerns about these approaches have also been raised. Few of the problems are as follows; varying complexities in the use case models, adjusting the technical complexity factors and experience factors, classification of use cases and the overall construction of the UCP method. Additionally, there are few problems associated with using Use Cases as well [1], [22]. First, there is no standardized style of writing a Use Case. The variations in the style and formality of writing a Use Case brings about many issues like how to measure the size of the Use Case, and how to classify the Use Case. Second, an important issue with Use cases is the assessment of complexity of the Use Case. In a typical CRUD (Create, Replace, Update, Delete), is it correct to consider the Use Case (UC) as one UC with four scenarios or one UC with one scenario, as all the other scenarios are so similar.

Third, a Use Case represents an external actor's view. In case the system has states, it becomes necessary to define another model to represent this behavior which is quite complex. Fourth, granularity of Use Cases is another big issue. What is the optimum length and what are the details that should be mentioned while describing a Use Case. Fifth, most of the researchers complain about the non-consideration of non-functional requirements in the Use Case models.

This raises the question that, are Use Cases a good choice to depend on for estimating effort? The answer lies with the proper evaluation and investigation of these approaches. Many proposed approaches have addressed these issues satisfactorily and many of them have ameliorated many problems as well. We discuss these approaches and compare them for analysis in the following sections.

3 Use Case-Based Metrics

In this section, we present a summary discussion of the effort estimation techniques we have selected for comparison. The summary has been presented to help the reader understand the basic idea of each effort estimation technique. The metrics to be compared are as follows:

- Use Case Points (UCP) [14]
- Transactions [24]
- Paths [24]
- Extended Use Case Points (EUCP) [30]
- UCPm [9]
- Adapted Use Case Points (AUCP) [18]
- Use Case Size Points (USP) [6]
- Fuzzy Use Case Size Points (FUSP) [6]
- Simplified Use Case Points (SUCP) [21]
- Industrial use of Use Case Points (IUCP) [7]

Use Case Points: The basic technique proposed by Gustav Karner [14] for estimating effort based on Use Cases. The method assigns quantitative weights to actors based on actor classification as simple, average and complex. The sum of all the weighted actors in the system gives the Unadjusted Actor Weight UAW. Similarly, Use Cases are classified according to their complexity and are assigned quantitative weights. The sum of all the Use Cases in the system gives the Unadjusted Use Case Weight UUCW. The sum of UAW and UUCW gives the Unadjusted Use Case Points UUCP. Then, a number of technical complexity factors and experience factors are weighted and are multiplied to the UUCP to yield Use Case Points UCP. Finally, the obtained Use Case Points are multiplied by the Productivity Factor PF to give the final Effort Estimate. Critics claim Karner's method to be decent with the exception of the non-flexibility in adjusting the Productivity Factor which was later proved to be a major variable affecting the estimation process.

Transactions: A metric proposed by Gabriela Robiolo *et al* [24] for estimating size of software based on the size of Use Cases. It depends on the textual description of a Use Case. A Transaction is defined by a stimulus by the Actor and response by the system. The sum of all the stimuli is the number of Transactions in a particular Use Case. Summing up the transactions for all the use cases in the entire system, the number of Transactions is calculated. In order to estimate the final effort, the Historical Mean Productivity technique was used by the authors [24]. Three major objectives using this metric and the following metric 'Paths' were highlighted by the method which are simplifying the counting method, to obtain different units of measurement that individually may capture a single key aspect of software applications and reducing the estimation error.

Paths: Another metric proposed by [24] which pursues similar objectives as the 'Transaction' metric. It is based on the concept of Cyclomatic complexity which identifies binary and multiple decisions in code. The same idea has been applied in terms of textual descriptions of Use Cases. The method is as follows; obtaining the complexity of each transaction. For obtaining the complexity of each transaction, first count the number of binary decisions, then identify the multiple decisions by counting the different pathways and subtract one from the number obtained. In the final step, for computing the complexity of each uses case, sum up the complexity value for each transaction.

Extended Use Case Points: The EUCP method proposed by Wang *et al* [30] contains three parts; first, refining the Use Case classification with fuzzy set theory. Second, using a learning Bayesian Belief Network BBN for getting the Unadjusted Use Case Points UUCP probability distribution. Third, using a BBN for generating the effort probability distribution which is derived from UCP. The contribution of this approach is a probabilistic cost estimation model obtained by integrating fuzzy set theory and Bayesian belief networks with the generic UCP method.

UCPm: A slight modification of the Use Case Points method proposed by Sergey Diev [9]. The method stresses more on defining Actors and Use Cases comprehensively. The slight change from the basic UCP method is the calculation of the size of the software product. The 'UUCP' obtained is multiplied with the technical complexity factor 'TCF' to give the size of the software product. To this, environmental factor 'EF', base system complexity factor 'BSC' and pre-defined number of person-hours per use case point 'R' are multiplied. Finally, supplementary effort factor is added to yield the final effort estimate of the software product. The supplementary effort may include activities like writing configuration management scripts or performing regression testing.

Adapted Use Case Points: The basic objective of this method proposed by Mohagheghi *et al* [18] is to develop a technique which fits the incremental model of software development and in situations where requirements specifications are frequently changed. The method follows the structure of the UCP method but with major differences. All actors are assumed to be average without differences in classification. All the Use Cases are assumed to be complex and then later on they are decomposed to smaller use cases and classified as simple or average. The method includes the extended use cases as well and counts them as base use cases. Exceptional flows are also counted as average use cases. The method has very promising results and the major contributions are the adaptation of the UCP method for incremental development and identifying the impact of effort distribution profile on effort estimation results.

Use Case Size Points: Proposed by Braz and Vergilio [6]. The metric focuses on the internal structures of the Use Cases in depth and hence better captures the functionality. The primary factors considered in this metric are the Actors classification, pre-condition classification and post-condition classification, main scenarios, alternate scenarios, exception classification and the Adjustment Factor. The sum of all these factors gives the Unadjusted Use Case Size Points UUSP which is subsequently multiplied by the difference of the technical complexity factor and the experience factor. The results are compared with Function Points and UCP metrics.

Fuzzy Use Case Size Points: Another metric proposed by Braz and Vergilio [6]. The primary factors considered in this metric are the Actors classification, pre-condition classification and post-condition classification, main scenarios, alternate scenarios, exception classification and the Adjustment Factor. The sum of all these factors gives the Unadjusted Use Case Size Points UUSP which is subsequently multiplied by the difference of the technical complexity factor and the experience factor. The difference between USP and FUSP is in the use of the concept of Fuzzification and

Defuzzification. This creates gradual classifications that better deal with uncertainty. Also, it reduces the human influence on the classification of the Use Case elements. The results obtained using this metric are slightly better than the Use Case Size Points metric.

Simplified Use Case Points: The main aim of this method proposed by M. Ochodek *et al* [21] is to simplify the UCP method and the process of Effort Estimation in general. This is not a completely defined metric. The approach used for realizing the objective is the cross validation procedure, which compares different variants of UCP with and without certain factors. Factor Analysis was also performed to investigate the possibility of reducing the adjustment factors. The results from this study include recommending a metric based on rejection of actor weights and rejection of 9 Technical Complexity Factors and 6 Experience Factors.

Industrial Use Case Points: The IUCP method proposed by Edward Caroll [7] is not a defined metric but an amalgamation of different industrial practices used in association with the UCP method to increase the accuracy and reliability of the estimation procedure. The main contribution of this method is the inclusion of the Risk Factor and additional effort for activities other than the development of the software product. Also, in depth analysis of few factors like Performance Analysis, Deliverable Analysis, Schedule Analysis, Defect Analysis, Causal Analysis and Quantitative Management Analysis is mentioned. The importance of using a Process Improvement Cycle is also highlighted.

4 Framework for Comparison

To compare the proposed metrics, we developed a framework consisting of ten attributes, which were chosen carefully to accommodate all the pros and cons of using those metrics. Unfortunately, there is no literature survey available in the specific domain of effort estimation based on Use Cases. As such, there are no previous evaluation attributes available. Nevertheless, few attributes have been borrowed from Saliu's and Ahmed's [26] work as well as Irfan's [1] work which was aimed at evaluating various size metrics. The qualified evaluation attributes and their descriptions are as follows:

Accuracy: The degree of precision or correctness obtained in estimating the effort with reference to a particular approach is termed as Accuracy. It is basically obtained by comparing the effort estimated with the actual effort and checking for deviations. A higher accuracy of an approach validates the efficiency of that approach. Better accuracy implies better reliability [1]. It should be noted that comparing estimation accuracy of various approaches is not easy pertaining to reasons such as different datasets, different definitions of similar terms and different goals of estimation accuracy [11].

Ease of Use: This attribute implies simplicity of use. How easy it is to use a particular technique/approach? A fact that should be understood is that, the effort required in estimating effort for software development should be minimal. What is the use of a technique which itself requires a lot of time and effort? [22]. Preferably,

the approach used should be simple enough to be implemented in a reasonable time frame as Bente Anda [3] states that the UCP method requires little technical insight and effort and hence makes it easy to use in early stages.

Use Case detail considerations: The level of detail considered in evaluating a particular Use Case before using it in the estimation process is important for various reasons. Issues like the granularity of Use Cases, number of scenarios in a Use Case, inclusion of Extended Use Cases with the Base Use Cases, classification of Use Cases as simple and complex are commonly debated among various researchers for the Use Case based estimation methods [9], [18], [28]. This is a valuable attribute for comparing the different approaches related to Use Case based methods.

Factor Inclusion: The effort estimation calculated using the basic UCP method considers various Experience factors and Technical Complexity factors [14]. The variety of other Use Case based approaches we have considered, discard few of these factors and consider them unrequired for the estimation process, whereas few of the approaches consider some additional factors [18], [21]. The attribute will help in analyzing the approaches and contribute in specifying the optimal factors to be considered in the estimation process.

Adaptability: The capability of the model or method to adjust according to new environments and fit the incremental style of development practices is termed as Adaptability of the model. “Incremental or evolutionary development approaches have become dominant. Requirements are changed in successive releases, working environments are shifted and this has been accepted as a core factor in software development” [18]. A method or a model should be adaptive to these changes and if it is otherwise, then the model will have limited usability value.

Handling Imprecision and Uncertainty: Quite a common aspect in all the software development practices is to take account of the imprecisions and uncertainty associated with the processes. We know that there is a reasonable imprecision in estimating the size of software and a lot of uncertainty in predicting various factors associated with developing software [19]. A model which considers these factors is better than a model which doesn't.

Sensitivity: The receptiveness or responsiveness to an input stimulus is called sensitivity. In terms of software development, a model in which the change in estimated effort with respect to a small change in the input values is large or significant is termed as a sensitive model. In Effort Estimation, it is desirable to have low sensitivity models.

Appropriate use of Productivity Factor: The conversion of estimated points based on Use Cases to Effort requires the multiplication of a factor called productivity factor whose units are person-hours. Initially, Karner [14] proposed a productivity factor value of 20 person-hours, which later turned out to be variable for different projects. An appropriate use of the productivity factor results in close to accurate estimations and reduces the deviations. This is a valuable attribute to distinguish between the available approaches.

Artifacts Considered: This attribute reflects the artifacts that are considered in the implementation of a particular technique or metric. Effort Estimation using Use Cases considers all the functional requirements in a satisfactory way, but a major complaint against the use of this method is that the non-functional requirements are not considered extensively. But, if the artifacts pertaining to non-functional requirements like estimating for reports, schedule spreadsheets, staffing concerns are considered [7], then the method could have a valid defense. The use of artifacts considered by different models is helpful in comparing them.

Empirical Validations: The evaluation and validation of a metric or a model in general is essential. If the model is validated, then the validation criteria and the dataset on which it is validated are considered. Datasets from the industry are considered more reliable than student datasets or datasets from open sources [1]. The empirical validation of a model adds to its credibility as well.

5 Comparison between the Metrics

This section presents the actual comparison and evaluation of the qualified metrics. It is worth noting here that we used subjective ratings in evaluating the different approaches. Future work will investigate applying more quantitative objective ratings. A point worth mentioning here is that, all the afore-mentioned metrics have been validated by using real time projects of large companies. The comparisons have been presented in tabulated form for sake of simplicity and ease of understanding. All the tables are followed by a short discussion which summarizes the tabulated information and provides recommendations for the use of certain metrics with respect to the attributes.

5.1 Accuracy

Metric	Comments
UCP[14]	Relatively good accuracy and promising results. More accurate than expert estimates in few cases and almost equally accurate in some other cases.
Transactions[24]	Good accuracy, close to UCP, lower variability of prediction error, high correlation with actual effort.
Paths[24]	Better accuracy than Transactions and UCP, lower deviation from actual effort, high correlation with actual effort.
EUCP[30]	Better accuracy than UCP as they use Fuzzification and a Bayesian Belief Network to train the system.
UCPm[9]	Relatively good accuracy, less calculations required in the method.
AUCP[18]	Very good accuracy, effort calculated using AUCP for release 1 and release 2 were 21% and 17% lower than Actual Effort.
USP[6]	Competent accuracy compared to others, but lower error rates.
FUSP[6]	Competent accuracy results with lower error rates, a fuzzified form of USP with minor changes in results.
SUCP[21]	Slight improvement in accuracy. Discarding TCF and EF doesn't cause a negative effect in prediction of effort.
IUCP[7]	Perhaps the most efficient and accurate results. Using the process improvement loop, the deviation in prediction has been cut down to 9%, which is a very significant contribution.

Discussion: Even after evaluating all metrics based on their respective results, terming a certain metric better than others is not justified because of many reasons such as different data sets used, differences in the nature of the software projects, environmental and expertise differences, etc. Nevertheless, it is recommendable to use metrics which use machine learning techniques like FUSP. Additionally, the use of industrial practices in the estimation process improves the accuracy of the method. Hence, the use of IUCP is also recommendable.

5.2 Ease of Use

Metric	Comments
UCP[14]	Very easy to compute effort using UCP. It can be done at the early stages of the development of the life cycle. A rough estimate can also be made just by mental calculation.
Transactions[24]	An easy method involving counting the number of transactions in each Use Case and subsequently the total in a system.
Paths[24]	A relatively complex method to use, involving obtaining the complexity of a transaction by summing up the number of binary decisions and identification and summing up of multiple decisions.
EUCP[30]	A complex method involving fuzzifying the inputs and training the Bayesian Belief Network for estimating effort and consequently defuzzifying the output to obtain a crisp value.
UCPm[9]	An easy method, almost similar to UCP; the only difference being size is calculated as the product of Unadjusted Use Case Weights and the sum of Technical Complexity factors.
AUCP[18]	A complex method compared to other approaches. Involves computing modified Unadjusted Use Case Weights and uses many additional factors such as Adaptation Adjustment Factor (AAF), and Equivalent Modification Factor (EMF) which itself comprises of 6 other factors.
USP[6]	A fairly simple method to calculate the effort. Only lengthy part is to consider the details of use cases and classify them appropriately.
FUSP[6]	A simple method, slightly complex than USP because of the Fuzzification of inputs and Defuzzification of outputs respectively.
SUCP[21]	A method simpler than conventional UCP, this reduces the number of Technical Complexity Factors and Experience Factors by limiting them to 6 only.
IUCP[7]	A simple method similar to UCP, with the additional overhead of calculating for non-functional requirements like documenting reports, spread sheets, etc.

Discussion: Almost all the metrics are subjectively rated equally in terms of ‘Ease of Use’, with the exception of Paths and AUCP metrics. It is intuitive that since the basic UCP method is quite simple in terms of use, a metric or method which deviates from the norms and structure of the basic method is bound to be relatively complex. Though the EUCP method is mentioned as complex, the rational can be to consider

the metrics which use soft computing methods as relatively more time consuming rather than terming them as complex to use. We recommend SUCP as the metric easiest to use compared to the others with UCP coming a close second.

5.3 Use Case Detail Considerations

Metric	Comments
UCP[14]	Only considers the complexity classification of a Use Case by counting the number of transactions in a Use Case. Classified as simple, average and complex.
Transactions[24]	Considers only the stimulus by an actor and response by the system, by counting the number of transactions. No other details are considered.
Paths[24]	Identifies binary and multiple decisions in a Use Case. Sums up the number of binary and multiple decisions in a Use Case and consequently for the entire system. No other details are considered.
EUCP[30]	The Use Case classification is refined by considering detailed aspects of a Use Case such as User Interface Screens, pre-conditions, primary scenario, alternative scenario, exception scenario, post-conditions.
UCPm[9]	High level of detail is considered. Scoping of actors, classification of Use Cases as zero weight use cases, duplicated use cases, background process use cases, report use cases. Also considers the granularity of use cases.
AUCP[18]	Initially all Use Cases as considered complex, then are broken down to simple and average based on transactions. Include extended Use Cases as base Use Cases and exceptional flows in a Use Case are also assigned a weight factor of 2.
USP[6]	A detailed classification comprising of pre-conditions, post-conditions, main scenarios, alternate scenarios and exceptional scenarios.
FUSP[6]	The Use Case detailed classification comprises of pre-conditions, post-conditions, main scenarios, alternate scenarios and exceptional scenarios.
SUCP[21]	Considers the complexity classification of a Use Case by counting the number of transactions in a Use Case. Additionally, the cardinality of Use Cases is computed.
IUCP[7]	Similar to UCP, IUCP does not consider any extra Use Case details except the complexity classification.

Discussion: This is perhaps a very important and valuable attribute for distinguishing the strengths and weaknesses of the available metrics. Majority of the metrics base their calculations of size on the number of transactions in a Use Case without considering other details related with use cases. If the metrics were to be ranked according to this attribute or recommended on this basis, Use Case Size Point ‘USP’ would win the evaluation followed by UCPm and AUCP. The reason for this ranking is quite visible in the tabulated information. USP considers almost all the details associated with a Use Case. UCPm takes it to a further level by classifying use cases by varying levels but misses including the pre-conditions and post-conditions.

5.4 Factor Inclusion

Metric	Comments
UCP[14]	Includes Actor weights and Use Case weights. Also includes 13 Technical Complexity Factors and 8 Experience Factors.
Transactions[24]	No use of Actor weights and Use Case weights. Does not include any Technical Complexity Factors and Experience Factors.
Paths[24]	No use of Actor weights and Use Case weights. Does not include any Technical Complexity Factors and Experience Factors.
EUCP[30]	Includes Actor weights, Use Case weights, 13 Technical Complexity Factors and 8 Experience Factors.
UCPm[9]	Includes Actor weights, Use Case weights, 13 Technical Complexity Factors, 8 Experience Factors. Additionally, UCPm includes Base System Complexity factor and Supplementary Effort Factor.
AUCP[18]	Actor Weights and Use Case weights are included. All the Technical Complexity Factors and Experience Factors are discarded. Includes new factors such as Adaptation Adjustment Factor (AAF), Equivalent Modification Factor (EMF), and Overhead Factor (OF).
USP[6]	Actor weights and Use Case weights are included as per the detailed Use Case classification. Additionally, 14 Technical Complexity factors and 5 Environmental Factors are included.
FUSP[6]	Actor weights and Use Case weights are included. 14 Technical Complexity Factors and 5 Environmental Factors are included.
SUCP[21]	Discards Actor weights and includes only Use Case weights. 9 out of 13 Technical Complexity factors and 6 out of 8 Experience Factors are discarded.
IUCP[7]	Includes Actor weights and Use Case weights. Also includes 13 Technical Complexity Factors and 8 Experience Factors.

Discussion: Perhaps the most debated attribute which can involve lot of future work. The issue is to find the optimum number of factors that are to be considered while estimating effort. Many metrics agree with the standardized thirteen technical complexity factors and the eight experience or environmental factors as proposed by the basic UCP method. SUCP discards nine technical complexity factors and six experience factors. UCPm keeps all the standard factors same but includes additional factors. Few metrics like Transactions, Paths and AUCP discard all the standardized factors but the latter makes up for the non-inclusion by using new factors such as AAF, EMF and OF. As such, we cannot recommend any metric to be the best in terms of this attribute.

5.5 Adaptability

Metric	Comments
UCP[14]	Very simple and adaptable method. Fits any Use Case modeling environment easily.
Transactions[24]	An adaptable method, worked well with 13 different projects under different environments. Fits the dynamic model of software development. Only needs counting the number of transactions.
Paths[24]	Fairly adaptable. Depends on calculating the complexity of Use cases. Slight difficulty expected in adapting to environments with less experienced teams.

EUCP[30]	Less adaptive as compared with other metrics because of the involvement of the training BBN.
UCPm[9]	Fairly adaptable to different environments. Difficulty with less experienced teams for estimating effort.
AUCP[18]	Perhaps the most adaptable metric. The aim of realizing this metric was to fit the incremental model of development and support environments where Extreme Programming is used.
USP[6]	Slightly less adaptable relatively. The adjustment factors need to be calibrated with each and every changing project and environment.
FUSP[6]	Same as the USP method. Less adaptable relatively.
SUCP[21]	Adaptable in many environments. Applied to 14 industrial and academia projects with relative ease and promising results were obtained. Removal of few factors supports adaptability.
IUCP[7]	A very adaptable metric, perhaps because of the feedback loop and its ability to fit into any mode of operation and environment. The metric has been custom designed to fit any model of development.

Discussion: Almost all metrics qualify well for this attribute. Few of them are more adaptable in terms of their structure, ease of use and lesser difficulty with new and inexperienced teams. An interesting observation is that, the use of soft computing methods like in the case of EUCP, where a learning Bayesian Belief Network is incorporated in the estimation process, it made the metric relatively less adaptable to different working environments. But the validity of this observation can be debatable. AUCP is the most recommended metric in terms of Adaptability.

5.6 Handling Imprecision and Uncertainty

Metric	Comments
UCP[14]	Doesn't handle imprecision, though it manages to deal with uncertainty up to some extent.
Transactions[24]	Doesn't handle imprecision nor uncertainty.
Paths[24]	It is not designed to handle imprecision and uncertainty.
EUCP[30]	Handles imprecision and uncertainty fairly because of the use of Fuzzy logic and additionally because of the learning Bayesian Belief Network.
UCPm[9]	Not capable of handling imprecision and uncertainty.
AUCP[18]	Does not handle imprecision, but the metric deals with uncertainty satisfactorily.
USP[6]	Is not capable of handling both imprecision and uncertainty.
FUSP[6]	The fuzzified version of USP, and hence it handles imprecision and uncertainty quite well.
SUCP[21]	Does not handle imprecision, nor does it handle uncertainty.
IUCP[7]	A metric tailored to deal with uncertainties but cannot handle imprecision.

Discussion: Another important factor for evaluation. It is much desirable that in a process like estimation of effort and cost where loads of uncertainty is possible and imprecise estimates are quite common, a metric should account for both the aforementioned factors. Unfortunately, most of the metrics don't account for both

imprecision and uncertainty. Few of them such as UCP, AUCP and IUCP are capable of dealing with uncertainties but not imprecision. EUCP and FUSP, since they use soft computing techniques account reasonably well for both imprecision and uncertainty and are recommended for use.

5.7 Sensitivity

Metric	Comments
UCP[14]	The metric is less sensitive to input changes. Can accommodate noise reasonably well.
Transactions[24]	Is less sensitive to changes. A small change to the input i.e. the increase or decrease in the number of transactions of a Use Case will not adversely impact the effort estimated.
Paths[24]	Is moderately sensitive when compared to Transactions metric. If the Use Case details are changed, the number of binary decisions and multiple decisions change considerably. This affects the final estimated effort.
EUCP[30]	Less sensitive because of the Fuzzification and Defuzzification process. Accommodates noise levels easily.
UCPm[9]	Less sensitive as the input factors don't impact the final estimated effort much.
AUCP[18]	A moderately sensitive metric. AUCP incorporates many factors because of which, a slight change in some factors may result in considerable changes to the final estimated effort.
USP[6]	Less sensitive to changes.
FUSP[6]	A slightly less sensitive metric than the USP. It accounts for varying levels of input changes.
SUCP[21]	A lesser sensitive metric. Almost similar to the conventional UCP metric.
IUCP[7]	Not sensitive to input changes. Works the dynamic way and hence accounts for changes anywhere in the process lifecycle.

Discussion: A much desirable attribute for comparison in many fields and not just effort estimation, Sensitivity like 'Use Case Details Consideration' can distinguish between metrics in a very proper way. Unfortunately, it is very difficult to distinguish between the available metrics because of lack of information related with the sensitiveness of the metric inputs and outputs. Nevertheless, few metrics have been classified as lowly sensitive and moderately sensitive. It is worth noting that, using soft computing approaches can minimize the sensitivity of a metric considerably. The IUCP can be recommended for use if Sensitivity is the main concern.

5.8 Appropriate Use of Productivity Factor

Metric	Comments
UCP[14]	Karner described the method and fixed the productivity factor at 20 man-hours per Use Case Point.
Transactions[24]	Effort calculation is based on Historical Mean productivity technique. No involvement of Productivity Factor.
Paths[24]	Effort Estimation is based on Historical Mean productivity technique. No involvement of Productivity Factor.

EUCP[30]	Not much use of the productivity factor. All the calculations are based on adjusting other factors.
UCPm[9]	Uses the productivity factor specified by the conventional UCP method.
AUCP[18]	Productivity factor of 36 man-hours per Use Case is used in addition to other adjustment factors such as AAF, EMF and OF. In case of the overhead factor (OF) not being used, the use of 72 man-hours as productivity factor has been prescribed.
USP[6]	A productivity factor of 26 man-hours is used as per the calculations.
FUSP[6]	Productivity factor of 26 man-hours has been used.
SUCP[21]	Productivity factor of 20 man-hours, 28 man-hours and 36 man-hours has been used as per the requirement of the project under consideration which is appropriate.
IUCP[7]	Productivity factor of 20 man-hours and 28 man-hours has been used as other adjustments are taken care of by the risk adjustment factor and factors like estimating for reports.

Discussion: With respect to Use Case based effort estimation, this attribute has a vital contribution in the comparative analysis. Earlier when the estimation of effort based on use cases was in its infancy, there were quite significant variations in estimated effort even though the technical complexity factors and experience factors were properly adjusted. The reason which came in the focus after many years was the inappropriate use of Productivity Factor. Since, Karner proposed a 20 person-hour per use case; it was not changed for quite some time until variations with it resulted in more accurate effort estimates. SUCP can be recommended for use as it allows variable use of the Productivity Factor with respect to the project. The use of IUCP is also recommended as it provides freedom to the estimators for selecting the appropriate Productivity Factor.

5.9 Artifacts Considered

Metric	Comments
UCP[14]	Does not take into account any additional artifacts.
Transactions[24]	Does not consider any additional artifacts. Deals with the functional requirements only.
Paths[24]	No consideration of additional artifacts.
EUCP[30]	No additional artifacts considered.
UCPm[9]	No additional artifacts are considered.
AUCP[18]	Considered artifacts related to non-functional requirements of the process lifecycle like availability, performance and security.
USP[6]	No consideration of additional artifacts.
FUSP[6]	No additional artifacts are considered.
SUCP[21]	Additional artifacts are not considered.
IUCP[7]	A lot many artifacts have been considered by the IUCP metric. Artifacts like estimating for reports, risk management artifacts, artifacts dealing with performance analysis, deliverable analysis, schedulable analysis and defect analysis are considered.

Discussion: In terms of this study, artifacts imply the inclusion of non-functional requirements in the effort estimation process. As tabulated in the above tables, most

of the metrics do not consider any additional artifacts with the exception of the AUCP and the IUCP. AUCP considers important non-functional requirements such as performance and security. IUCP also considers non-functional requirements in addition to including lesser effect artifacts such as Reports documentation etc. As such, both AUCP and IUCP are recommended for use.

5.10 Empirical Validations

Metric	Comments
UCP[14]	Many empirical validations are available for the use of traditional UCP approach. Many authors have validated the UCP procedure empirically using both Industry datasets as well as Student datasets.
Transactions[24]	Empirically validated using datasets comprising of 13 small business projects distributed across 3 different contexts; an Undergraduate Academic Environment, System and Technology Department at Austral University and a level 4 CMM certified company. The projects are also distributed implementation wise as well.
Paths[24]	The same datasets used to validate the Transactions metric were used.
EUCP[30]	Validated using two industry projects in a Chinese company of 500 employees. Since results show some inconsistency, more evaluation needs to be done with the metric.
UCPm[9]	Not validated using any dataset. The proposed metric is a result of analysis carried out over 50 projects in a period of 2 years as reported.
AUCP[18]	The results of applying this metric were validated using a telecom project of Ericsson and across 2 releases. The authors report more case studies that validated the AUCP metric but information about them has not been specified explicitly.
USP[6]	A case study was done to validate the results of this metric using a real project database of a private company. The metric was validated against Function Points and traditional UCP.
FUSP[6]	Same case study as was used by the USP metric. FUSP was validated against Function Points, traditional UCP and USP itself. Differences between USP and FUSP were also highlighted. The use of these metric needs more validations and more experiments needs to be done.
SUCP[21]	Empirically validated against 7 industrial projects and 7 other projects from the Poznan University of Technology. The range of the actual effort was 277 man-hours to 3593 man-hours. Promising results were obtained. Additionally, a framework was built to evaluate the estimation accuracy of all the 14 projects using this metric.
IUCP[7]	The metric has been validated over a continuous period of 5 years, consisting of 200 projects in a CMM level 5 company. The results are astonishing as the feedback loop helped in reaching 9% deviation with reference to the Actual Effort for 95% of the company's projects.

Discussion: The attribute where in all the metrics are on par with each other. It is interesting to note that all the metrics have been extensively validated using Industrial data sets. As such, we cannot underestimate the evaluations of the proposed metrics in any manner.

6 Analysis

Based on the critical survey and after drawing comparisons between the various Use Case based metrics on a common ground, several shortcomings arose which were anticipated. The comparison brought forth many weak links in the Use Case based estimation process and at the same time highlighted many advantages of using it.

Nearly all the metrics have been validated extensively using industry datasets and student datasets. This is an onus for the validity of the efficiency and accuracy of the metrics. This is well complemented by the fact that most of them have competent and reliable effort estimates. Most of the proposed metrics are easy to use which makes them more liable to be favored over other techniques and metrics which provide similar results. Adaptability, in terms of usage of the metrics is noteworthy considering that almost all metrics qualify as being fairly adaptable and the case studies involving them verify the fact. Few metrics consider detail classification of the Use Cases with respect to complexity by considering all the aspects related to the implementation of Use Case. Metrics which capture the details are definitely more useful and efficient than metrics which do not consider detailed classification. Also, the inclusion and exclusion of the technical complexity factors and experience factors showed varied results. Mostly, it was generalized that the exclusion of few factors does not have negative impact on the estimation of effort. Many metrics considered the technical complexity factors to be overlapped and hence discarded many such factors.

Sensitivity is an attribute which could not be properly addressed in the comparison. It is due to the fact that enough information was not available to distinguish the metrics from being highly sensitive and lowly sensitive. It is desirable to have metrics and techniques which have low level of sensitivity. Based on our comparison, few metrics were found to be lowly sensitive and few moderately sensitive. Productivity factor is an important concern while estimating effort using Use Cases. It is an important contributor for the conversion of the metric in terms of size to effort. Appropriate use of this factor affects the final estimated results. The degree of correlation between estimated effort and Actual effort can be established satisfactorily if the productivity factor is rightly used. Most of the proposed approaches don't consider the importance of this factor and focus more on other adjustment factors. Use of expert opinion or analogy can be used to at least appropriately select the Productivity factor.

The two most important and perhaps the negative factors in terms of using Use Case based metrics are the inability to deal with imprecision and uncertainty and little consideration of additional non-functional artifacts. These two attributes show the vulnerability of the Use Case based approach when compared with other approaches. Most of the compared metrics do not account for imprecision with the exception of the metrics using Fuzzy logic and other machine learning techniques. Uncertainty, however, did not seem to have caught enough attention; future research is needed to consider the uncertainty associated with measurements provided by the different metrics.

One of the most important weaknesses of Use Case based approaches was the non-consideration of the non-functional requirements associated with software development processes. Though few metrics attempted to incorporate the artifacts

pertaining to non-functional requirements, it is not enough. Any software process depends on both functional and non-functional requirements. A metric or technique which does not consider additional artifacts will have varying levels of deviation in the estimated effort.

Despite few shortcomings and negative aspects, the detailed comparison and evaluations support the fact that estimating effort using Use Cases is justified and that they can be successfully used in the software effort estimation process. The important requirement is that the negative aspects which expose the vulnerability of Use Cases should be addressed. In the same context, if a standardized approach is established to write Use Cases, many issues would be minimized. Alternately, each organization can come up with their own standards of writing Use Cases and keep a check on the standards so that, the estimation process can be generalized using Use Cases. The incorporation of non-functional requirements is an essential paradigm that should be taken care. It would remove lot of pessimism about the reliability and efficiency of the use of Use Case metrics. Lastly, using the process improvement lifecycle as a feedback loop to learn and incorporate efficient techniques should be prescribed by organizations so as to reap the benefits of efficient and accurate effort estimation. Causal Analyses and Quantitative Management Analysis of the reports documented should be carried out on a periodic interval to ensure continuous improvement.

7 Conclusion and Future Work

Estimating effort in software development is a difficult and challenging activity. There is no metric or technique which can be preferred over other techniques in all cases and circumstances. Each technique has its corresponding advantages and disadvantages. Nevertheless the focus should be on developing metrics and techniques which complement the desired capabilities. Due to its early applicability during the development lifecycle, use case based metrics have gained wide acceptance recently and have been proven to yield promising results. More research should be dedicated to develop metrics to overcome the negative aspects discussed above though.

Moreover, the variety of use case-based size metrics, which has been proposed, suggests that there may be some inconsistencies among the measurements computed using these metrics. In turn, such inter-inconsistencies raise the concern that relying on measurements of one single metric might not lead to the same estimation of the effort. An obvious important candidate for future work is to study the uncertainty that should be considered when relying on a single metric. The uncertainty arises due to the inability of the metric designer to comprehensively consider all factors that would indeed contribute to the use case size as a predictor for effort; that is neglecting some factors due to the lack of a complete theory of the concept of size and effort, the impracticality to list all the factors that affect the size and effort, etc. In other words, future work should research a framework meant to facilitate portraying the probability distribution of the error associated with measurements computed using a given metric. This, in turn, allows associating a *degree of reliability* to the effort estimated by a given metric; that is a level of how dependable such estimate is.

Even though the evaluation attributes were carefully selected, there may certainly be some additional attributes which can help in better evaluation from a different perspective. Attributes like Sensitivity could not be properly addressed because of lack of insufficient information in the corresponding metrics description. It is much desirable to distinguish between metrics as being lowly sensitive and highly sensitive. Moreover, as pointed out earlier, the use of subjective ratings in evaluating the different metrics need more clarity; future work will investigate applying more quantitative objective ratings. This would help in recommending a particular metric as the best metric in terms of practical use for software practitioners and software developers.

An important work would be to address the problem of use cases not accommodating non-functional requirements. This is very important in terms of effort estimation as the consideration of non-functional requirements can bring about reasonably large variances in the estimated effort. Typically, in the industry, the common practice to avoid this problem is by including supplementary effort which includes effort pertaining to the non-functional requirements. But this is quite vague. A detailed work like the concept of *misuse cases* for eliciting security requirements by Guttorm Sindre and Andreas Opdahl [27] should be carried out for including the non-functional requirements in a software project.

Another direction in which work needs to be done is to understand whether the inclusion or exclusion of the Technical Complexity factors and Experience Factors brings forth any significant differences in the estimated Effort. In the Adapted Use Case Points [18] approach, all the technical complexity factors and experience factors are excluded and factors like the Adaptation Adjustment factor, Equivalent Modification factor and the Overhead Factor are included. The change in effort as a result of this factor exchange is portrayed to be better compared to the previous approach. In [21], authors have discarded nine technical complexity factors and six experience factors terming those factors as ‘not required’ in the estimation process. No other techniques have recommended this factor minimization. Analysis needs to be done to understand the effect of factor inclusion in the estimation process.

An interesting observation is the dearth of usage of machine learning techniques in the estimation approaches based on Use Cases. With the exception of Use Case based estimation, many an approach in Effort Estimation utilizes the benefits of the machine learning techniques. Interestingly, there is no work in the literature which uses soft computing in the estimation process. An effort to incorporate fuzzy logic in the estimation process has been attempted by [6] and [30]. Future work in this area is of paramount importance especially given the benefits of using soft computing.

Acknowledgements. The authors wish to acknowledge King Fahd University of Petroleum and Minerals (KFUPM) for utilizing the various facilities in carrying out this research.

References

- [1] Ahmad, I.: A Probabilistic Size Proxy for Software Effort Estimation: A Framework, Master Thesis, Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia (April 2008)
- [2] Albrecht, A.J., Gaffney, J.E.: Software function, source lines of codes, and development effort prediction: a software science validation. *IEEE Trans. Software Eng.* SE-9, 639–648 (1983)
- [3] Anda, B., Benestad, H.C., Hove, S.E.: A multiple case study of Effort Estimation based on Use Case Points. *Empirical Software Engineering* (2005)
- [4] Boehm, B.: *Software Engineering Economics*. Prentice-Hall, Englewood Cliffs (1981) ISBN 0-13-822122-7
- [5] Boehm, B., Abts, C., Chulani, S.: *Software Development Cost Estimation Approaches: A Survey*, University of Southern California Centre for Software Engineering, Technical Report, USC-CSE-2000-505 (2000)
- [6] Braz, M.R., Vergilio, S.R.: Software Effort Estimation based on Use Cases. In: *Proceedings of the 30th Annual International Computer Software and Applications Conference (COMPSAC 2006)*, IEEE, Los Alamitos (2006)
- [7] Carroll, E.R.: Estimating Software based on Use Case Points. In: *OOPSLA 2005*, October 16-20, ACM, New York (2005) 1-59593-193-7/05/0010
- [8] Costagliola, G., Ferrucci, F., Tortora, G., Vitiello, G.: Class Point: An Approach for the Size Estimation of Object-Oriented Systems. *IEEE Transactions on Software Engineering* 31(1), 52–74 (2005)
- [9] Diev, S.: Use Cases modeling and software estimation: Applying Use Case Points. *ACM Software Engineering Notes* (November 2006)
- [10] Forbes, M.: *Use Case Survey. Towards Adopting Enterprise Standards for Use Cases* (2009)
- [11] Grimstad, S., Jorgensen, M.: A Framework for the Analysis of Software Cost Estimation Accuracy. In: *Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering*, pp. 58–65.
- [12] Hareton, L., Zhang, F.: *Software Cost Estimation*, Department of Computing, Hong Kong Polytechnic University,
<http://paginaspersonales.deusto.es/cortazar/doctorado/articulos/leung-andbook.pdf>
- [13] Jacobson, I., Booch, G., Rumbaugh, J.: *The Unified Software Development Process*. Addison Wesley, Reading (1999)
- [14] Karner, G.: *Metrics for Objectory*. Diploma thesis, University of Linkoping, Sweden. No. LiTH-IDA-EX-9344:21 (December 1993)
- [15] Kirsopp, C., Shepperd, M.J., Hart, J.: Search heuristics, case-based reasoning and software project effort prediction. In: *Genetic and Evolutionary Computing Conference (GECCO 2002)*. AAAI, New York (2002)
- [16] Lai, R., Huang, S.J.: A Model for Estimating the Size of a Formal Communication Protocol Specification and Its Implementation. *IEEE Transactions on Software Engineering* 29(1), 46–62 (2003)
- [17] MacDonell, S.G., Gray, A.R.: A Comparison of Modeling Techniques for Software Development Effort Prediction. In: *Proceedings of the 1997 International Conference on Neural Information Processing and Intelligent Information Systems*, Denedin, Newzealand, pp. 869–872. Springer, Heidelberg (1997)

- [18] Mohagheghi, P., Anda, B., Conradi, R.: Effort Estimation of Use Cases for Incremental Large-Scale Software Development. In: ICSE 2005, May 15-21, pp. 1–58113. ACM, New York (2005) 1-58113-963-2/05/0005
- [19] Muzaffar, S.Z.: Adaptive Fuzzy Logic based Framework for handling imprecision and uncertainty in software development Effort prediction models, Master Thesis, Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia (January 2006)
- [20] Neill, C.J., Laplante, P.A.: Requirements Engineering: The State of the Practice. *Software* 20(6), 40–46 (2003)
- [21] Ochodek, M., Nawrocki, J., Kwarcia, K.: Simplifying effort estimation based on Use Case Points. *Journal of Information and Software Technology*, 0950-5849 (2010)
- [22] Probasco, L.: Dear Dr. Use Case: What about Function Points and Use Cases?, <http://www.ibm.com/developerworks/rational/library/2870.html>
- [23] Ribu, K.: Estimating Object-Oriented Software Projects with Use Cases, Master of Science Thesis, Department of Informatics, University of Oslo, Oslo, Norway, November 7 (2001)
- [24] Robiolo, G., Badano, C., Orosco, R.: Transactions and Paths: two use case based metrics which improve early effort estimation. *IEEE, Los Alamitos* (2009) 978-1-4244-4841-8/09
- [25] Robiolo, G., Orosco, R.: Employing use cases to early estimate effort with simpler metrics. *Innovations System Software Engineering* 4, 31–43 (2008)
- [26] Saliu, M.O., Ahmed, M.A.: Soft Computing Based Effort Prediction Systems – A Survey. In: Damiani, E., Jain, L.C., Madravio, M. (eds.) *Soft Computing in Software Engineering*. Springer, Heidelberg (2004) ISBN 3-540-22030-5
- [27] Sindre, G., Opdahl, A.: Eliciting security requirements with misuse cases. *Requirements Engineering* 10, 34–44 (2005)
- [28] Smith, J.: The estimation of effort based on use cases. *IBM Rational Software White Paper* (1999)
- [29] Strike, K., El-Emam, K., Madhavji, M.: Software Cost Estimation with Incomplete Data. *IEEE Transactions on Software Engineering* 27(10) (October 2001)
- [30] Wang, F., Yang, X., Zhu, X., Chen, L.: Extended Use Case Points Method for Software Cost Estimation. *IEEE, Los Alamitos* (2009) 978-1-4244-4507-0/09

An Agent-Based Autonomous Controller for Traffic Management

Sadia Afsar¹, Abdul Mateen¹, and Fahim Arif²

¹ International Islamic University,
Islamabad, Pakistan

² Computer Science Department, Military College of Signals, National
University of Sciences & Technology,
Islamabad, Pakistan

sadiaa.afsar@gmail.com, abdulmateen@fuuastisb.edu.pk,
fahim@mcs.edu.pk

Abstract. Emerging trends in software development has been changed due to the huge amount of data, growth of internet, mobile, dynamic and smart applications. These applications consist of small, intelligent, flexible and distributed components known as agents. This research proposes agent-based autonomous controller (ABAC) architecture for managing road traffic. It uses time series of historical traffic intensity to estimate the appropriate time allocation for each signal at a given intersection. Our approach takes care of the exceptional appearance of rescue vehicles (e.g., ambulance) in order to ensure a smooth flow of the traffic. The ABAC architecture counts on several AI techniques germane to assessing the intensity of the traffic using image recognition algorithms. It also counts on environment sensors (sound sensors) in order to detect the advent of emergency vehicles. The ABAC traffic management architecture shows a high degree of adaptability leading to the least need for human intervention.

Keywords: Traffic Management, Autonomous, Controller, Agent, Prediction.

1 Introduction

Roads, vehicles and traffic signals are the most important players in any traffic management system. Roads are highways that intersect each other at some specific point. Vehicles are the moving objects such as cars, buses, etc. Traffic signals are the central points where the vehicles can change or remain in the same direction. A vehicle at any given signal can move in three directions, left, right and straight [1]. The vehicles on a signal are controlled through three lights or a single light having three states i.e. red, green and yellow. Signal light changes according to four rules, i.e. green to yellow, red to yellow, yellow to green and yellow to red as shown in Fig. 1. There is no direct conversion of lights from green to red and vice versa.

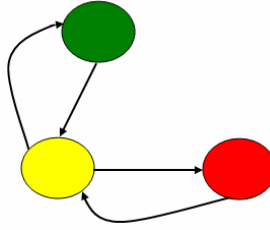


Fig. 1. Traffic Signal Light Sequence

Traffic jam (Fig. 2) is a common problem around the world and it incessantly imposes challenge on researchers to find effective solutions. At a given intersection point, an effective traffic management should give sufficient time for each vehicle to pass without causing deadlock or backlog.

Use of computers in traffic control (air, ships, and vehicle) system has been increased due to the powerful computation of processor-based systems. It also becomes essential as population and vehicles growth is rapidly rising. All these factors throw in to automate the traffic control system. The automation manages the traffic control itself without the involvement of human beings; which result in reducing the workload with avoiding congestion and increases the traffic flow. Number of systems has been automated through agent oriented and autonomic computing approach. IBM introduced the concept of autonomic computing in 2001 [1]. The aim of this technology is to develop such systems that perform action and manage itself without any external input or support from the humans. Basic characteristics of autonomic computing are self-optimization, self-inspection, self-configuration, self-healing, self- organization and self-protection [1, 2]. Concept of artificial intelligence and agents programs are introduced by Turing in 1950 [11]. Turing gave the idea that how intelligence can be incorporated in machines. He also proposed that how learning machine can be built and how teaching ability can be integrated. Agents have some special characteristics such as mobility, autonomous, adoption, goal oriented and interaction with other agents or systems to do some useful tasks.

The agent technology is actually an extension of component based technology with more interactive and dynamic nature [14]. Agents are self-contained and independent systems that perform numerous actions in order to achieve predefined tasks. In agent-oriented systems, the agents work collaboratively to solve the problem at hand. One of the advantages of this technology is the parallel execution of numerous number of agents on a single machine.

At many occasions, analytical models may not show flexibility towards the solution as their underlying math is intractable. Such models are often deterministic as they do not show adaptive behavior towards the dynamics of the parameters of the problem. In this paper, we present a management framework for the road traffic, ABAC (Agent-Based Autonomous Controller), that aims at allocating appropriate time window for each signal at a given intersection. It also handles the special cases

of emergency vehicles. The proposed architecture is adaptive and employs well known Artificial Intelligence techniques germane to image processing in order to count vehicles in a given image.

In this piece of work, we are proposing an agent-based architecture that is autonomous with respect to gleaning data from the environment using sensors and making decisions that should handle the traffic loads smoothly in order to mitigate congestions. The rest of the paper is organized as follows. Section 2 describes the previous work on traffic management architectures. Section 3 presents our proposed architecture to handle traffic management problem while section 4 concludes the paper with some future directions.



Fig. 2. Traffic jam is a common problem especially in big cities

2 Related Work

Much research has been done in the context of autonomic computing and agent oriented methodologies [7-13]. In the context of traffic management, a number of systems has been developed that adopt the autonomous and agent technology. For example, Tavlidakis et. al. [12] presented a technique that controls the traffic in a distributed fashion. Their traffic management is performed by placing the controllers at intersection points only. These controllers collaborate by exchanging data among themselves. The system uses GIS principles and acts autonomously in the sense that it can control the traffic for isolated nodes and towns. Nevertheless, these remote controllers run under the direct supervision of human being at the command centre that is located at some central position.

Lee et. al. [13] proposed a Traffic Parameter Measure Algorithm (TPMA) to derive traffic parameters from video image sequences. The algorithm analyzes images and observes vehicle movement to extract the traffic parameters such as vehicle quantity, speed, etc. The proposed algorithm does not require any special image processing hardware and can work on a simple PC. TPMA analyzes frames based on different parameters such as brightness, noise, and many others in combination. The algorithm was validated via empirical experiments and found that the manual and proposed algorithm converges to the same count of vehicles.

Alagar et. al. [7] discussed a model for autonomous traffic control system that uses a decentralized approach where a separate controller for each lane is used to dispatch the traffic information to an arbiter. The arbiter has to allocate time slots such that the

road conditions remain collision free. There are twelve such sub-systems and one arbiter. The said technique is described by visual and formal description in Unified Modeling Language (UML) where class, state chart and collaboration diagrams depict the proposed system for road traffic system.

Thangiah et. al. [8] discussed the vehicle routing problems (VRP) and suggested the solution through agent architecture. Their proposed architecture consisted of distributed omni search strategy that solved the variants of the VRP problems without changing the actual system. Moreover, this architecture is featured by the ability of solving complex variants without rewriting the algorithms. The presented solution taps into well-known AI algorithms like genetic algorithms, simulated annealing and neighborhood search strategies.

Hewage et. al. [16] optimized the timing of traffic signal through a special purpose simulation tool. This tool has the capability to optimize the timing of signal light at single as well as multiple junctions. The tool requires less resources and the traffic administrator who even do not know about the simulation can draw road network. He simply positions the road network through icons and can draw junctions with entering the actual traffic demand. The tool provides city network and results in visual form. Administrator can understand and analyze outputs through statistical data that are based on simulation. However it is important that actual results can only be obtained if the input data is correct. The simulation tool for traffic management is developed in Symphony 1.05.

Liu et. al. [17] used multi-agent architecture to design an intelligent transport system (ITS), which provides information about the road emergency or other unexpected event to drivers which help to make correct decisions. ITS mainly consists of information processing application and road condition transferring module. Information processing application module is responsible to collect analyze traffic data, while road condition transferring module is to make reaction according to the road condition information. There are two agents in ITS, which are Manager agent and Vehicle agent. Manager agent creates or deletes the Vehicle agent, collect and distribute messages. The Vehicle agent explore GIS and GPS, produces maintenance graph, select best path, check emergency areas and send this information to Manager agent. For communication they used different wireless technologies such as ad hoc network, GPS, GIS, WIMAX and WiFi.

Our research is different from the previous work in a number of ways. No previous technique for automated traffic control and management handles the emergency or rescue vehicles. Our proposed architecture detects the rescue vehicles and sends this information to next signals in advance so that vehicle can be able to pass each signal smoothly. The introduced approach uses hybrid architecture, i.e. centralized and decentralized, with intelligent and autonomic capabilities. This technique is centralized as the time allocation for all four signals of an intersection is controlled by a single controller (Observer agent) while the internal architecture of each controller operates in a de-centralized manner using sub-agents. The other distinction from the previous works is the use of dynamic time allocation for each side of the intersection, which ensures adaptation to the road conditions.

3 The ABAC Traffic Management Architecture

Traffic control problem can be solved by using the autonomous agents. In this paper, we introduce the *ABAC* (Agent-Based Autonomous Controllers) architecture for traffic management. In this architecture, each component of traffic control architecture is represented by an autonomous agent capable to take decisions and actions on their own according to desire and requirements. The abstract view of the proposed ABAC architecture is shown in Fig 3. It consists of supporting/ environmental components, knowledge base and observer agent. Supporting components include cameras, sound sensors and actuators while the observer agent consists of the sub-agents which are Decision Making agent, Analyzer agent, and the Learner agent.

The ABAC architecture collects traffic data using sensors, cameras or sonic sensors, and sends this information to the analyzer agent. The analyzer processes the image and counts the number of vehicles on each side of the intersection and sends this information to the decision making agent. The analyzer agent uses image processing techniques to count the current traffic on each side. In case of receiving an alert about a rescue vehicle from the sensor, the decision-making agent stores the current state of signals in the knowledge base, closes all traffic signals except the one from which the emergency vehicle is approaching. After passing the rescue vehicle, the previous state of signals and their opening sequence gets retrieved from the knowledge base. In the non-emergency cases, the decision-making agent analyzes the counted vehicles of each side and allocates threshold for a signal. The threshold gets updated every hour after analyzing the traffic data that will be retrieved from the knowledge base. After the allocation of threshold, the time of each signal will be allocated according to the traffic on particular side. The same procedure will be repeated after the signal threshold. The above description represents the steps of ABAC architecture at an abstract level. In the coming section, we are discussing the detailed view of the ABAC architecture as shown in Fig. 4.

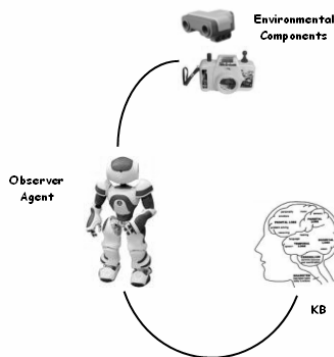


Fig. 3. Abstract view of the ABAC Architecture

The ABAC architecture collects traffic data using sensors, cameras or sonic sensors, and sends this information to the analyzer agent. The analyzer processes the

image and counts the number of vehicles on each side of the intersection and sends this information to the decision making agent. The analyzer agent uses image processing techniques to count the current traffic on each side. In case of receiving an alert about a rescue vehicle from the sensor, the decision-making agent stores the current state of signals in the knowledge base, closes all traffic signals except the one from which the emergency vehicle is approaching. After passing the rescue vehicle, the previous state of signals and their opening sequence gets retrieved from the knowledge base. In the non-emergency cases, the decision-making agent analyzes the counted vehicles of each side and allocates threshold for a signal. The threshold gets updated every hour after analyzing the traffic data that will be retrieved from the knowledge base. After the allocation of threshold, the time of each signal will be allocated according to the traffic on particular side. The same procedure will be repeated after the signal threshold. The above description represents the steps of ABAC architecture at an abstract level. In the coming section, we are discussing the detailed view of the ABAC architecture as shown in Fig. 4.

3.1 Environmental Components

1) *Sensor*: The sensor in ABAC architecture is used for monitoring which can be a camera and a sonic detector. The camera at each side is used to capture the image of the incoming traffic flow while the sound sensor is required to detect the emergency, rescue or other specific alarms. The captured information of image and/ or sound is communicated to the Analyzer agent for further analysis and processing.

The ABAC adopts the following workflow: System takes the input in the form of images from camera and sound alert from the sound detector.

Case1: Image Captured from Camera:

The image on each side will be taken 10 seconds before the completion of signal cycle time or threshold. After taking the image, it is processed by counting the number of vehicles on each side/ image. On the basis of total vehicles; the decision is made whether to continue the current signal or switch to the next one. The waiting time of each vehicle depends upon the traffic flow of all four sides and signal status.

Case2: Sound Detected by Sonic Sensors:

When the rescue, emergency or other some special sound is detected through the sonic sensor, the analyzer agent stops all other working and activities; and sends this important information to the DM agent that will provide service to rescue or other such type of vehicles in a proper manner. The sound detection algorithm [20] will work by storing the audio sound of all rescue vehicles and assign index to each sound. When a new sound is detected, it will be matched with stored voices and if it is matched with of the stored voice then this information is sent to DM agent.

2) *Actuators*: The actuators switched the traffic signals that may be yellow, green or red. The actuators will work as an execution component and are controlled through the DM agent.

3.2 Observer

The main function of observer agent is to regulate the traffic flow. It not only synchronizes its internal agents but also to synchronize with environmental

components. The observer agent allocates the proper amount of time and provide collision free environment. The observer agent consists of three sub-agents, Analyzer, Decision Making (DM) and Learner agent. These sub-agents of the Observer Agent are described as under:

i. *Analyzer Agent*: The analyzer agent is responsible for taking input from the sensor which consists of image and/ or sound with direction. Incoming image will be preprocessed by removing the noise and irrelevant data. The analyzer agent counts the number of vehicles on each side of the signal by using the Background Subtraction algorithm [18, 19]. This algorithm consists of four steps which are preprocessing, background modelling, foreground detection and finally data validation. The processing step take the image and removes noise by spital, temporal smoothing and morphological processing. The background modeling step is used to describe the statistical status of the backgeound. The foreground detection step recognizes those pixels of the image that were not identified in background modelling step. The output (candidate foreground mask) of this step is in a binary form. At the end, data validation is performed by eliminating the pixels that are not a part of moving objects and generates foreground mask. Number of vehicles counted on each side of the signal is sent to the DM agent. If the incoming data from the sensor has also rescue alarms/ sounds then Analyzer agnet stops all other processing and sends this information to the DM agent for quick response. The emergency and rescue vehicles can also be detected from pre-processed image. However, we are using sound sensor approach, thaty is more beneficial as the rescue vehicle may be far away or may be hidden behind other vehicles in image.

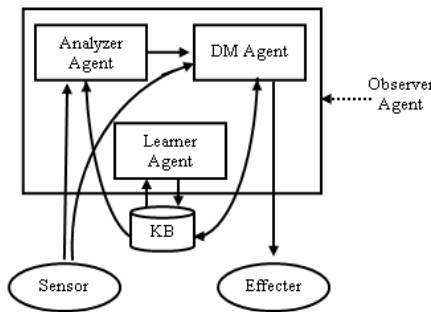


Fig. 4. Architecture of the ABAC System

ii. *Decision Making Agent*: The DM agent can receive input from the Analyzer component. In case of sound input from the sonic sensor, the DM agent will turn the opened signal to red after a while (10 secs); store the current sequence into the knowledge base; and turn the signal green from where rescue vehicle is coming, the DM agent will close the current signal and once again procedure continues to previous sequence that is stored in the Knowledge Base. DM agent will send this information about the emergency or rescue vehicles to the Observer agent of the next signal, which arrange the situation for the incoming emergency vehicle. In case of input from the analyzer agent; the DM agent compares the number of vehicles with

each other and threshold. If the number of vehicles exceeds then it will check the order from the Knowledge Base so that next signal could be opened. Threshold will be set automatically by taking the first time image of each side.

iii. *Learner Agent*: The learner agent takes the data from the knowledge base, identifies useful patterns and on the basis of these patterns improves the knowledge base. The improved knowledge will be used by the analyzer and DM agent to take future decisions more intelligently. Learner agent improves the accuracy for arrival of the emergency vehicle from one signal to its successor.

3.3 Knowledge Base

The Knowledge base is a main repository for storing different types of relevant information; that includes the threshold of each side of the signal, distance from the nearest signals from all its four sides, opening order of a signal and the information about rescue alarms. All the agents use knowledge base to retrieve and store the information for further processing.

4 Conclusion and Future Work

The research proposes an autonomous agent-oriented traffic control system to manage and defuses congestions on roads. The proposed solution uses a hybrid approach where the traffic from each side is controlled through a single controller while the internal details are taken care of by autonomous agents. The proposed architecture controls the traffic with minimum human involvement. Furthermore, it is featured by the special treatment for the rescue vehicles. The architecture is to a large extent adaptive to the road conditions especially to the intensity of traffic stream. Additionally, the ABAC architecture can be integrated with other systems such as traffic bulletin boards to share traffic information with the public. The proposed solution for the traffic management will be evaluated through a case study in future that will reveal the performance of the ABAC architecture over some well known previous traffic management architectures. This case study will be based on the different traffic scenarios and the way by which our proposed architecture handles them efficiently.

References

1. Mueller, E.A.: Aspects of the history of traffic signals. *IEEE Trans on Vehicular Technology* 19(1), 6–17 (1970)
2. Horn, P.: *Autonomic Computing: IBM's Perspective on the State of Information Technology*, IBM Journal Paper (2001)
3. Kephart, J.O., Chess, D.M.: The Vision of Autonomic Computing. *Computer* 36(1), 41–50 (2003)
4. An Architectural blueprint for autonomic computing, IBM White paper, 3rd edn. (2005)
5. Huebscher, M.C., Mccann, J.A.: A survey of Autonomic Computing Degrees, Models, & Applications. *ACM Computing Surveys* 40(3) (2008)

6. Stuart, R., Peter, N.: *Artificial Intelligence, A Modern Approach*, 2nd edn. (2008) ISBN No. 81-7758-367-0
7. Alagar, V.S., Muthiayen, D.: *A Rigorous Approach to Modeling Autonomous Traffic Control Systems*. In: *The Sixth International Symposium on Autonomous Decentralized Systems (ISADS)*, Italy, pp. 193–200 (2003)
8. Sam, R., Olena, T., Mennell, W.: *An Agent Architecture for Vehicle Routing Problems*. In: *SAC*. ACM, New York (2001)
9. Pour, G.: *Expanding the Possibilities for Enterprise Computing: Multi-Agent Autonomic Computing*. In: *10th IEEE International EDOCW 2006*, pp. 33–33 (2006)
10. Tian, J., Tianfield, H.: *A Multi-agent Approach to the Design of an E-medicine System*. Springer, Heidelberg (2003)
11. Bernon, C., Capera, D., Mano, J.-P.: *Engineering Self-Modeling Systems: Application to Biology*. In: *Int. Workshop on Engineering Societies in Agents World. LNCS*. Springer, Heidelberg (2008)
12. Tavladakakis, K., Voulgaris, N.C.: *Development of an autonomous adaptive traffic control system*. In: *The European Symposium on Intelligent Techniques*, Greece, June 3-4 (1999)
13. Yu, H.L., Yu, T.L.: *A fast algorithm for measuring traffic vehicle parameters*. In: *Proceedings of the 7th International Conference on Machine Learning & Cybernetics*, Kunming, pp. 3061-66 (2008)
14. Luck, M., McBurney, P., Preist, C.: *Agent Technology: Enabling Next Generation Computing* (2003)
15. van Aart, C.: *Organizational Principles for Multi-Agent Architectures*. Series: WSSAT – Whitestein Series in Software Agent Technologies (2005) ISBN 3-7643-7213-2
16. Kasun, N., Hewage, J.Y.: *Ruwanpura: Optimization of Traffic Signal Light Timing Using Simulation*. In: *Winter Simulation Conference*, pp. 1428–1433 (2004)
17. Liu, X., Fang, Z.: *An Agent-Based Intelligent Transport System*. In: *11th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 304–315 (2007)
18. Cheung, S.-C., Kamath, C.: *Robust Background Subtraction with Foreground Validation for Urban Traffic Video*. *EURASIP Journal on Applied Signal Processing* 14, 1–11 (2005)
19. Cheung, S.-C., Kamath, C.: *Robust Techniques For Background Subtraction In UrbanTraffic Video*. In: *Video Communications and Image Processing*. SPIE Electronic Imaging, San Jose (2004)
20. Casey, M.: *MPEG-7 Sound Recognition Tools*, Mitsubishi Electric Research Labs, Cambridge, MA, Unites States of America

Comparative Evaluation of Performance Assessment and Modeling Method for Software Architecture

M.A. Isa and Dayang N.A. Jawawi

Faculty of Computer Science and Information Systems
Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia
{mohdadham, dayang}@utm.my
<http://www.utm.my>

Abstract. Conducting performance assessment during the early phases of system development enhances early design decisions of system design. The generated performance models from system design will help in identifying the potential performance problems in the system design based on the result of the performance assessment. In recent years, several methods for performance assessment and modeling have been proposed. This paper presents the comparative evaluation of performance assessment and modeling methods to discover the strengths and weaknesses of the existing methods based on a set of criteria which includes process and modeling elements that was developed with the purpose to represent a specific process to assess the performance attributes with the help of modeling concepts. The four selected methods were evaluated based on these criterions and the results will hopefully guidance for developers to propose better methods for performance assessment and modeling in the future.

Keywords: Performance Assessment, Performance Model, Model Transformation, Software Performance Engineering, Model Driven Engineering, Unified Modeling Language (UML), Model Evaluation.

1 Introduction

The process of performance assessment and modeling in system architecture is useful in identifying the important modules that require detailed model specification, performance requirements estimation and target performance models prediction. Performance is generally a quality attribute that meets the functionality of the system with timelessness and correctness [1]. In addition, the ISO9126 standard defines performance as efficiency that requires two main factors: time behavior and resource efficiency [1]. These factors usually address common performance metrics such as response time, throughput and utilization. Most applications use these metrics for performance assessment, which leads to the importance of performance modeling in order to address performance metrics parameter correctly in systems model.

Performance assessment aims to evaluate the capabilities of knowledge or methods towards a given tasks or action. For example, given a set of system functionalities, the assessment is conducted to evaluate the performance of each systems functionality with a pair of performance objectives. Performance objectives may include a performance metrics with a specific threshold such as response time. System functionality should meet a minimal level of according to the performance metrics otherwise the evaluated system functionality is considered underperformance. In another context, software architecture is the structure of the system that consists of the systems component with specific behavior that triggers the communication between components [2]. Performance assessment should be done on the software architecture to ensure the correctness of the system structure and response towards a given scenario. To support the assessment tasks, performance modeling is a key artifact that enables the transition from analysis on to systems design. Nowadays, modeling language such as the UML MARTE [3] profile has been used to address performance aspects in various domains. Large sets of performance annotations have been developed to enrich systems design annotation with performance aspects. With this, systems model with rich performance annotation can be transformed into performance model by using model-based performance prediction approaches [4]. Draw this insight together, performance assessment and modeling is possible for system development methodology to address performance aspects as early as possible in the system development lifecycle.

The aim of this paper is to conduct a comparative evaluation for performance assessment and modeling method. The motivation behind this is to explore and compare how existing methods conduct performance assessment and modeling in various domain and what is the value of model for representing systems architecture and performance model. In dealing with this issue, the selected criterion for the comparative evaluation is provided. Each method is then evaluated towards the given evaluation criteria in order to classify the methods with systematic evaluation.

The rest of the paper is organized as follows: Section 2 reviews the methods for performance assessment and modeling and the evaluation criteria selection is expressed in Section 3. The result of the comparative evaluation is highlighted in Section 4. In section 5, the comparative evaluation result is discussed. Section 6 concludes this paper with the expected future works.

2 Method for Performance Assessment and Modeling

The quality of the system is highly influenced by the correctness of the system functionality and quality attributes. System quality attributes such as performance can ensure the completeness of the system since performance requirements have a huge impact on the systems architecture, which in consequently, affects the system operation. In respect to this, early performance assessment for system architecture is highly desirable to prevent underperformance during system deployment. A specific method is needed to assess or evaluate the performance of the system architecture with the support of a modeling language.

This section describes the performance assessment and modeling method for system architecture where the methods include Performance Assessment of Software Architecture (PASA) [5], Continuous Performance Assessment of Software Architecture (CPASA) [6], Performance Refinement and Evolution Model (PREM) [7], and Performance By Unified Model Analysis (PUMA) [8]. These methods were selected as they provide complete lifecycle for executing performance evaluation, which comprises of process and modeling approach.

2.1 Performance Assessment of Software Architecture (PASA)

PASA [5] is a method that is highly influenced by the SPE [9] methodology for conducting performance assessment on system architecture. An early evaluation on system architecture is essential to discover the quality attributes especially in the aspect of performance. PASA is intended to handle the performance problems in a system architecture that might occur during the analysis process. PASA generally consists of ten steps, each with varying performance characteristics. The performance characteristics are obtained from systems scenario since PASA is highly influenced by scenario-based approach.

In addition, the PASA method, which consists of ten steps, can be divided into four prominent areas. The area includes overview, identification, analysis and refinement. As for the overview area, the process and architecture of the system will be assessed in order to gain high-level understanding within the development team. After the understanding of the system architecture is gained and documented, a critical scenario related to performance is identified and each respective critical scenario will be bonded with the identified performance objectives. With the performance objectives and selected scenario, performance models are then developed and analyzed to discover potential performance problems. A set of performance problems is produced and a possible solution is proposed to rectify those problems. The result from solving system architecture problem in terms of performance in PASA can enhance the robustness of the system besides decreasing the risk of system failure during deployment.

2.2 Continuous Performance Assessment of Software Architecture (CPASA)

CPASA [6] method is proposed as an enhancement of the PASA method for enhancing performance assessment of software architecture. Originally, the PASA method aims to assess the performance aspects of system architecture with consideration of the specifications and requirements of the system when it is already completed. In response to this, CPASA was designed to enable continuous change as early as the requirements engineering level. System requirements evolve throughout the development lifecycle in order to obtain the best final design decisions. Therefore, any performance problems could be detected and fixed along the way to avoid problems in the final design decision.

The CPASA method revises the original PASA method in two main aspects. The first aspect is the development scopes. Unlike the PASA method, which focuses on the design steps, the CPASA method covers the requirement step until the design step in terms of performance assessment. The second aspect is the minimization of step in the performance assessment process. The CPASA method concentrates on minimizing the performance assessment steps with the help of the agile development process. This process intends to benefit the stakeholders when the requirements are frequently changed during the performance assessment process. The result from solving performance issues in system architecture with the CPASA method can provide a complete system design with the continuous change in the initial design decision during development process.

2.3 Performance Refinement and Evolution Model (PREM)

The PREM [7] method is designed to systematically manage the system performance aspects from the requirements phase until the testing phase. The main priority of this method is to enable requirements refinement and evolution. Refinement means that the system requirements can be refined during system development in order to obtain concrete requirements. A set of system requirements are able to evolve with more performance related aspects. In addition, the PREM method uses quantitative requirements and workload specification for managing system performance aspects.

The PREM method, which aims to manage system performance, based on quantitative requirements and workload specification is realized into four level processes. At each level, performance objectives are identified and a corresponding performance model is developed in order to analyze and test which level of workloads are affected for each performance scenario. Furthermore, the result for each level is tested before moving on to the next level to ensure the goal meets the initial criteria.

2.4 Performance by Unified Model Analysis (PUMA)

The PUMA [8] method is developed to bridge the gap between design model and analysis model in terms of performance attributes. The design model is designed using a well-known UML language with the support of performance annotation and, will be transformed into analysis model (performance). In between the transformation, possible problems could arise because of the different model semantics between the systems model and the performance model. The PUMA method deals with this issue by addressing a common intermediate model called Core Scenario Model (CSM)[10]. CSM is a meta-model that is based on the UML SPT profile in the performance context. CSM consists of multiple UML behavior diagram with the support of performance annotation and resource information, which later will support the development of different kinds of performance model.

The PUMA method consists of several steps where each step correlates the performance context for systems model. The system is modeled with the support of performance annotation, which is obtained from the UML SPT [11] or

MARTE profile. The model is then transformed into the intermediate model, the CSM, before it is converted into a performance model. The performance model is analyzed to produce the performance results and based on this, possible performance problems can be discovered and refined at the design phase. The result of solving performance context using the intermediate meta-model in PUMA has shown to close the gap of model semantics between the systems model based in the UML modeling language and the performance model.

3 Evaluation Criteria

This section briefly describes a criterion used for performance assessment and modeling evaluation. The approach for measuring each criterion was based on the Feature Analysis Approach [12]. This approach aims to classify a degree of criterion in two ways: simple form and scale form. A simple form means a yes/no argument form, which describes the probability of existence response for each criterion elements. On the other hand, the scale form aims to classifying the degree of conformance of the criterion against a feature. Since a method for performance assessment and modeling focuses on the process of development and modeling used for system and performance models, this criterion is used to evaluate the corresponding criteria based on scale and simple form. The set of criterion was selected through intensive reviews of performance-related methods specifically in the development process and modeling approach. The set of criterions were divided in two set:

Process related criteria. used to investigate what is the typical development process for performance assessment. This set of criterion validates whether the method follows all the basic process, covers the maturity of the process, specifies a target domain and enhance an existing software engineering approaches [13].

Modeling related criteria. used to evaluate how the system and performance models are developed. This criterion was selected based on the evaluation of model elements in the software architecture [14-15]. This set of criterion covers the performance related issues such as modeling views, modeling evaluation approaches, models transformation, and meta-model.

Each set of evaluation criteria was assessed through related inner elements, which influence the description of those criteria. As for the process related criteria, we defined a common description for the development phases. This includes the generic system lifecycle, performance activities and maturity stage. Generic lifecycles aim to define the traditional system development phases, which includes requirement engineering, analysis, design, implementation and testing. In addition, a performance activity describes an assessment of performance requirements during system development with the help of workloads identification. Workload identification is very important for performance assessment and modeling as it draws out the constraints and resources needed for the system [16].

Furthermore, domain analysis is also considered in the development process and methodology type as it describes which existing software engineering approach is being enhanced for each method. The detail of the process related criteria is shown in Table 1.

For the second evaluation criteria, modeling the performance models as a response from the systems model was essential in order to draw out the performance related requirements for the system. A respective inner element for modeling the related criteria is composed into modeling views, evaluation approaches, and transformation and meta-modeling. These elements enable necessary information to be extracted in order to support the system and performance models in the system development. The details of the modeling related a criterion is shown in Table 2.

Table 1 and Table 2 show the evaluation criteria for performance assessment and modeling method. As mentioned in the previous section, a selected method was evaluated with the defined criterions and the results could provide better understanding about the performance assessment and modeling method. In addition, these criterions can be used as basis for researchers to develop a new method through discovering the shortcomings of the existing methods.

4 Analysis Results of Comparative Evaluation

This section describes the comparative evaluation result of the selected methods against the given evaluation criteria.

4.1 The Process Related Criteria

The main priority for comparative evaluation in the context of process related criteria is to find out what is the generic development process followed by the methods. During the development process, the maturity of the models is traced with the support of performance activities.

The result showed that most of the methods scored either high or average for generic lifecycles especially at requirements, analysis and design level considering the need for early performance assessment at the earliest possible phases of development compared to the implementation and test level. As for the performance activities, most of the methods score high in the assessments and workloads element except for the PUMA method, which scored low for the workloads elements. The PUMA method is not really put on constraint for workloads because this method does not consider resource usage and distribution during the development phases.

In terms of domain applications, PASA and CPASA are specifically for real time systems while PREM and PUMA are intended to support general-purpose domains. Furthermore, the foundation of each method is different and was mostly inspired by the existing software engineering technology. The PASA, CPASA and PREM methods are highly motivated by the SPE methodology, which focuses on performance domain. Unlike the other method, PUMA uses the CSM meta-model as the intermediate model in the model transformation process. CSM is

Table 1. Process Related Criteria

Criterion		Description of level
Generic lifecycle	Requirements Engineering	Low: The method does not provides coverage for the phase Average: The method provides guidelines for the phase High: The method provides detailed guidelines for the phase
	Analysis	
	Design	
	Implementation	
	Test	
Performance Activities	Assessment	Low: The method does not address any guidelines for the activity Average: The method provides general description for the phase High: The method provides detailed description for the activity
	Workloads	
Domain		A: The methodology does cover general domain B: The methodology does cover specific domain
Methodology Type		A: The methodology has been developed from scratch B: The methodology has extending certain Software Engineering technology (MDE, MDA, SPE, CBSE)
Maturity Stage		A: The methodology has not address the level of maturity (inception, refinement) B: The methodology explicitly support the level of maturity (inception, refinement)

Table 2. Modeling Related Criteria

Criterion			Description of level
Modeling Views	System	Structure	<p>Low: The method does not provides coverage for the model elements</p> <p>Average: The method provides general description for the model elements</p> <p>High: The method provides detailed presentation for the model elements</p>
		Behavior	
	Performance	UML-Based	
		Analytical	
		Numerical	
Modeling Evaluation Approaches	Model-Based		<p>A: The model evaluation approach does not addresses in development phase</p> <p>B: The model evaluation approach is addressed in development phase</p>
	Simulation		
	Measurement		
Model Transformation	Same model		A: High B: Low
	Different model		
	Traceability between model		
Metamodeling Support			Does the method explicitly provide a metamodel for the performance modeling guides?

a meta-model that was created based on the UML SPT meta-model to express the performance formalism of performance models. The PUMA method uses this method to extract behavior related model from the UML annotated model. In addition, a criterion for model refinement for the systems model maturity is essential for both early and final design decision. All methods provide refinement for the systems model to acquire the best model before implementation except for the PREM method. This is because this method does not stress on the modeling part as describe in the previous section. The details of the results are shown in Table 3.

4.2 The Modeling Related Criteria

In order to deal with the large scale system specifications which includes the performance requirements and system functionalities, there is a need to model a system specification. Through models, developers are visible to illustrate and analyze the system specifications for performance assessment.

The result of the modeling criteria for the selected methods is shown in Table 4. As mentioned in the previous section, there are four primary criteria related to the modeling dimensions, each of them, composed into inner elements criteria. From the evaluation result, all the methods have modeled the structure and behavior of the systems. Typically, only the PUMA method scored high for systems structure model definition while the other methods score low. Except for the PREM method, all methods likely scored high in the behavior model definition. Performance model definition can have a profound effect on the system functionalities. Therefore, all methods have addressed the model based performance model whether in formal form (UML based) or traditional form (analytical) in contrast to the numerical form.

Continuing the model related criteria, for modeling evaluation approach criteria, all methods with the model-based approach consider the model as the primary artifact in representing the performance model. The PASA and CPASA methods added the simulation approach in the modeling evaluation while the other methods ignored this approach. In regard to this, all the methods exclude the measurement approach as the approach is applied after the implementation phase.

Model transformation between the systems architecture and the performance model is essential to define whether the performance model is semantically the same or different from the source model. Despite the fact that all the methods are less motivated to perform model transformation, all the methods except the PREM method included transformation of a model to the performance model but with different modeling languages. As for traceability between models, the PREM and PUMA methods enable the refinement process to enhance the models maturity during system development.

Furthermore, meta-modeling support is another criterion to guide developers in modeling performance model. Only the PUMA and PREM methods have defined the meta-model for performance model, while the CPASA and PASA methods ignore this criterion. The PREM method provides a general meta-model

Table 3. Comparative evaluation results for process related criteria

Method		PASA	CPASA	PREM	PUMA
Generic life-cycle	Requirements Engineering	Low	High	High	Low
	Analysis	High	High	Average	High
	Design	High	High	Average	High
	Implementation	Low	Average	Low	Low
	Test	Low	High	High	Low
Performance Activities	Assessment	High	High	High	High
	Workloads	High	High	High	Low
Domain		B	B	A	A
Methodology Type		Adopted SPE	Adopted PASA and enhanced with Agile development	SPE	Core Scenario Model (CSM)
Maturity Stage		B	B	A	B

as the modeling guideline without considering any UML rules such as MOF. In contrast, the PUMA method solely follows the UML modeling language with the support of the CSM meta-model for model transformation. The CSM meta-model is entirely for scenario-based and stresses on the use of resource in each step of the activities.

5 Discussion

The comparison of the evaluation of performance assessment and modeling methods was defined here to discover the possible gaps between the methods accompanied with the different views of evaluation criteria. The selected evaluation criteria was based on the assessment process and modeling approach. These criteria were selected because they represent the whole performance assessment process and the artifacts included in each process.

From the results of the evaluation, all methods explicitly addressed the process of conducting performance assessment for software architecture in terms of process views. The CPASA method provides better assessment process compared to other methods in terms of generic system lifecycle. Enhanced from the PASA method, the CPASA method is highly influenced by the agile development process. The Agile development process provides better refinement and increases the maturity of system specifications and designs throughout system development lifecycle. Compared to the other methods, this proves as the main advantage of the CPASA method.

In terms of the modeling views, the PUMA method has shown a strong confidence in modeling the systems model and the performance model. Unlike other

Table 4. Comparative evaluation result for modeling related criteria

Criteria		PASA	CPASA	PREM	PUMA	
Modeling Views	System	Structure	Low	Low	Low	High
		Behavior	High	High	Low	High
	Performance	UML-Based	Average	High	Average	High
		Analytical	Average	Average	Average	High
		Numerical	Low	Low	Low	Low
Modeling Evaluation Approaches	Model-Based		B	B	B	B
	Simulation		B	B	A	A
	Measurement		A	A	A	A
Model Transformation	Same model		B	B	B	B
	Different model		A	A	B	A
	Traceability between model		B	B	A	A
Metamodeling Support		N/A	N/A	Yes	Yes	

methods, which do not have strong focus on the systems modeling, PUMA addresses a systematic modeling for systems with the support of UML specification and annotated with the UML SPT/MARTE profile for the performance annotation. In addition, PUMA also provides systematic model transformation by adopting the CSM meta-model in bridging the gap between the systems model and the performance model as both models have different model semantic. In terms of design feedback, the PUMA method enables the iteration process to be applied for the whole process. However, it lacks focus on this aspect since its primary focus is in model transformation. Compared to the other methods, none of these methods provide systematic model transformation. The summary of comparative evaluation is shown in Figure 1.

Issues raised by the results from the comparative evaluation led to the requirement for additional focus on the modeling views for the performance assessment methodology. Modeling views are essential during the development as a model is a method to represent the model abstraction, which is close to the real world. As shown in Table 4, the modeling views, especially the model transformation features, should be given more attention. Model transformation could have a big impact in performance assessment for software architecture since it enables the close relationship between the design model and the analysis model. In addition, the transformation between different models semantic could later lead to problems in terms of complexity. The introduction of an intermediate model may solve this problem since an intermediate model would enable performance formalism for the performance models.

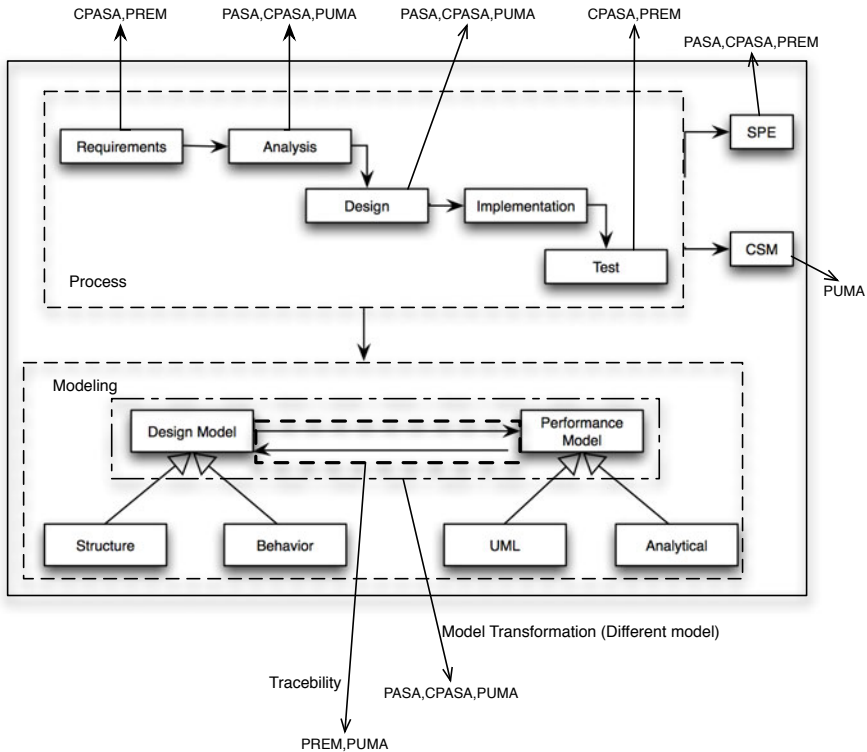


Fig. 1. Summary of Comparative Evaluation

6 Conclusion and Future Work

This paper provided comparative evaluation of performance assessment and modeling methods specifically in the software architecture level. The methods for performance assessment were selected and evaluated with a number of evaluation criteria. The criteria for evaluation purposes were divided into two major categories; process related and modeling related. Each criterion was selected to discover which standard process was executed by those methods with the support of modeling languages for the modeling of systems models and performance models. Researchers could use the result of this paper as a form of guidance in developing new performance assessment methods for software architecture.

The long-term goal of this research is to explore modeling related criteria, which could accommodate the use of MDE [17] in formalizing the modeling techniques used for model transformation. Model transformation could be a crucial part in transforming a systems model into a performance model. Hence, model transformation techniques should be taken into account the direct transformation of model with different model semantics could lead to complexity in the model transformation process.

Acknowledgments. The authors would like to thank Software Engineering Department staff and the members of Embedded Real-time System Software

Engineering Lab (ERETSEL), Faculty of Computer Science and Information Systems (FSKSM), UTM for their support.

References

1. ISO9126-2. Software engineering - Product quality - Part 2: External Metrics. ISO/IEC. Engineering Structures 31 (June 2001), doi: 10.1016/j.engstruct.2009.05.007.(2001)
2. Antoniou, G., van Harmelen, F.: A Semantic Web Primer. The MIT Press, Cambridge (2004)
3. UML Profile for MARTE : Modeling and Analysis of Real-Time Embedded Systems (2009)
4. Balsamo, S., Di Marco, A., Inverardi, P., Simeoni, M.: Model-based performance prediction in software development: a survey. IEEE Transactions on Software Engineering 30(5), 295–310 (2004), doi:10.1109/TSE.2004.9
5. Williams, L.G., Lane, R., Smith, C.U.: PASA SM: A Method for the Performance Assessment of Software Architectures. Architecture, 179–189 (2002)
6. Pooley, R.J., Abdullatif, A.A.L.: CPASA: Continuous Performance Assessment of Software Architecture. In: 2010 17th IEEE International Conference and Workshops on Engineering of Computer Based Systems, pp. 79–87 (2010)
7. Ho, C.W., Williams, L.: Deriving Performance Requirements and Test Cases with the Performance Refinement and Evolution Model (PREM). Evolution
8. Woodside, M., Petriu, D.C., Petriu, D.B., Shen, H., Israr, T., Merseguer, J.: Performance by unified model analysis (PUMA). In: Proceedings of the 5th international workshop on Software and performance - WOSP 2005, pp. 1–12. ACM Press, New York (2005), doi:10.1145/1071021.1071022
9. Smith, C.: Introduction to software performance engineering: Origins and outstanding problems. In: Bernardo, M., Hillston, J. (eds.) SFM 2007. LNCS, vol. 4486, pp. 395–428. Springer, Heidelberg (2007)
10. Petriu, D.B., Woodside, M.: An intermediate metamodel with scenarios and resources for generating performance models from UML designs. Software & Systems Modeling 6(2), 163–184 (2006)
11. OMG. Uml profile for schedulability, performance and time specification (2002), <http://www.omg.org/technology/documents/formal/schedulability.htm>
12. Kitchenham, B., Linkman, S., Law, D.: DESMET: a methodology for evaluating software engineering methods and tools. Computing and Control Engineering Journal 8, 120–126 (1997)
13. Asadi, M., Ramsin, R.: MDA-Based Methodologies: An Analytical Survey. In: Model Driven Architecture Foundations and Applications, pp. 419–431. Springer, Heidelberg (2010)
14. Ali-Babar, M., Zhu, L., Jeffery, R.: A framework for classifying and comparing software architecture evaluation methods. In: Proceedings of 2004 Australian Software Engineering Conference, pp. 309–318 (2004)
15. Mohagheghi, P.: An Approach for Empirical Evaluation of Model-Driven Engineering in Multiple Dimensions. sintef.com, <http://sintef.com/upload/IKT/9012/ECMFA10-C2M-Evaluating-MDE-final.pdf> (retrieved February 24, 2011)
16. Downey, A.B.: The elusive goal of workload characterization. ACM SIGMETRICS Performance Evaluation Review 26(4), 14–29 (1999)
17. Schmidt, D.C.: Model-driven engineering. IEEE computer 39(2), 25–31 (2006)

A New Approach Based on Honeybee to Improve Intrusion Detection System Using Neural Network and Bees Algorithm

Ghassan Ahmed Ali and Aman Jantan

Universiti Sains Malaysia

Ghassan@cs.usm.my, Aman@cs.usm.my

Abstract. A new approach inspired by bees' defensive behaviour in nature is proposed to improve Intrusion Detection System (IDS). In honeybee colonies, guards discriminate nestmates from non-nestmates at a hive entrance using an approach contains Undesirable-Absent (UA) or Desirable-Present (DP), and Filtering Decision (FD) methods. These methods are used to detect intruder and classify its type. In the proposed approach, the UA detector is responsible for detecting pre-defined attacks based on their attack signatures. Neural network trained by Bees Algorithm (BA) was used to learn the patterns of attacks given in training dataset and use these patterns to find specific attacks in test dataset. The DP detector is responsible for detecting anomalous behaviours based on the trained normal behaviour model. Finally, FD method is used to train the UA detector in real-time to detect new intrusions. The performance of the proposed IDS is evaluated by using KDD'99 dataset, the benchmark dataset used by IDS researchers. The experiments show that the proposed approach is applied successfully and able to detect many different types of intrusions, while maintaining a low false positive rate.

Keywords: Intrusion detection system; honeybee approach; neural networks; bees algorithm.

1 Introduction

Researches in computer security technologies remain obsession for many years of improvement and growth. However, it still needs a lot of hard work to settle the critical security problems. According to the 2010 Cyber Security Watch Survey [1], the number of security incidents continues to increase faster than companies' defenses. Within the reports, the outsider attacks are the main threats of cybercrime in general. However, more costly incidents are caused by the insider.

As a result, certain techniques are used to secure data, such as firewall, encryption etc. Nevertheless, most defense systems are still susceptible to attacks and intrusions. Therefore, the need of the detecting system that is able to detect the intrusion attempts from attacking the whole system is a very critical issue. The intrusion detection system (IDS) aims to support the essential security issues via scrutinizing every entry

and then the feedback for the user regarding the system situation. It acts as the "second line of defence" inside the network, giving a clear picture of threats that a system faces.

The concept of intrusion detection was born with Anderson's paper in 1980 [2]. Since then, several researches have been published to improve IDS to its current state. The main problem that arises when deploying IDS is that the deployment tends to generate excessive numbers of false alarms. In addition to that, IDS fails, in most cases, to meet high detection rate. These two cases are because of the difficulty that IDS faces to determine whether such action is either a malicious or a normal. Also, should IDS generates an alarm or not on such instance. Both of these weaknesses significantly reduce benefits of IDS and make the area of IDS open and attractive to researchers to solve these problems.

Many researchers such as [3] and [4] have argued that social insects' behavior system provides us with a powerful metaphor that can be applied to IDS problems. The ability to recognize and detect the intrusion is critical to the maintenance of the integrity of the social insect colonies [3]. In the proposed approach, we lean on the honeybee in nature, which faces the analogous security problems. Honeybees survive in difficult environments with different levels of threats to security. These threats motivate the bees to be able to detect and respond quickly on any aggressive acts that may attack the colony [5].

The problem faced by the honeybee guard is the same as the one faced by IDS, which is to distinguish between the intruder and the nest-mate. Honeybee colony has a small entrance which is patrolled by its' workers called guards [6] who allow nest-mates and deter the intruders. The entrance guards intercept and examine incomers at the nest entrance [6] and differentiate between nestmates and non-nestmates. The two methods Undesirable-Absent (UA) and Desirable-Present (DP) that the HoneybeeGuard uses in filtering the incomer are applied to the IDS. Further details about these methods will be indicated in Section 3.

In order to take the advantages of the proposed approach, we think it needs a technique to implement the idea effectively. One of the important requirements of this technique is the ability of learning. Moreover, this technique is supposed to distinguish between different characteristics after some level of training. Thus the neural network has been chosen to be the main component of the model. Neural network has many features such as the ability of learning, generalizing attributes even with noisy data, and the capability of classifying patterns effectively. These features can be further used to improve detection and reduce false alerts in the IDS.

Nevertheless, neural network alone cannot take the complete advantages of the approach because it has some drawbacks such as a computational complexity, slow of learning process, and difficulty of parameter settings. All of these problems will result poor performance of the system detection. Therefore many global optimization techniques have been proposed to train neural network to tackle these problems and enhance learning efficiency such as Particle Swarm Optimization [4] and Genetic Algorithms [7].

In this paper, an alternative method based on the Bees Algorithm (BA) [8] is proposed to be used in training neural networks under our approach. BA is a new optimization algorithm that imitates the natural foraging behavior of bees. It proposed

by [8] and has been successfully applied to different optimization problems including the training of neural networks for wood defect identification [9] and control chart pattern recognition [10] and show better results than other methods. To the best of our knowledge, the construction of training neural network by the bees algorithm to improve the performance of intrusion detection system has not been addressed in the literature. Here, this algorithm extends to classification and demonstrates its effectiveness in intrusion detection.

2 Background

2.1 Security Aspects of Honeybee in Nature

The similarities between a computer security's problems and the ones encountered by the honeybees can be seen by interpreting the honeybee colony behaviour into a computer security conduct. The security aspects, Figure 1, are emphasized in the honeybee colony behaviour; confidentiality and integrity have high priority in the colony, special guard bees scrutinize every entering individual and elicit a colony defense when non-nest members try to intrude [11]. Guards protect their nest from various robbers including the bees from other colonies.

Availability means enabling any nest member to access the nest at any time and to use the resources with cognizable rights. Accountability means examining other bees not only at the nest entrance but also inside the nest [6]. The early-warning system which the honeybee used to detect threats and clarify the intruder makes the nest system always safe. Correctness means participating in the nest defence as many bees from comb rush to join defence group to defend their colony from the intruders [12]. The multilayer protection in honeybee colony and the diversity of defences can be viewed as a typical framework of the detection system. Detection is essential in maintaining the integrity in the honeybee colony as well as in the IDS. Indeed, the similarity of the problem has continued to match up in the deeper levels with the computer security [13].

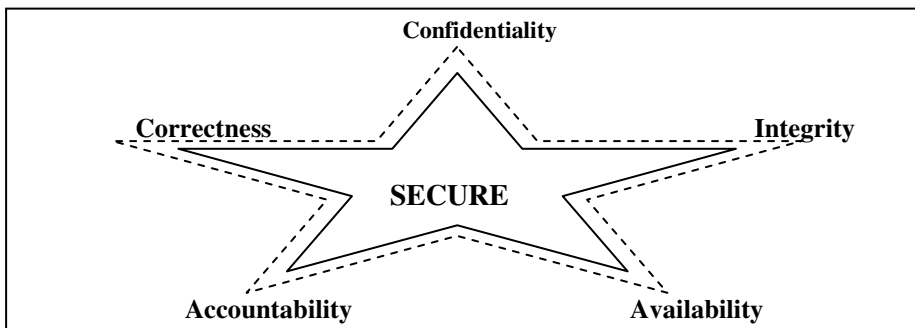


Fig. 1. Computer Security Aspects

2.2 Neural Network

A neural network [14] is a set of simple units called neurons working in unison to solve specific problems. It is inspired by the way biological nervous systems, such as the brain, process information. Signals can be passed between neurons through a series of weighted connections. Neural networks, like people, learn by example. When an input is presented, the network estimates an output. An adjustment of the weights between units results a function that capable to compute an appropriate output. This adjustment can be provided by an external teacher - (like Bees algorithm) - or by self supervised. After some modification rule, the networks become closer to the desired output.

A neural network is configured for a specific application, such as pattern recognition or data classification, through a learning process and it gives several advantages of using neural networks in the field of classification. Thus the neural network will be used as a component of the approach for improving an intrusion detection system. Relatively, Ryan et al. [14] have applied neural networks to learn the users print (command used) and report any behaviour which did not match the user print as abnormal behaviour. The neural network intrusion detection (NNID) system was tested and trained on limited number of users on offline mode. According to the authors, the system has the ability to work on large network environment. However, several experiments carried out to test the performance of the system and found that NNID is ideal to learn users' behaviour based on the users print on offline mode.

A hybrid approach using artificial neural network and fuzzy clustering (FC-ANN) proposed by [15] to improve the detection precision for low-frequent attacks and detection stability. The FC-ANN firstly divides the training data into several subsets using fuzzy clustering technique. Subsequently, it trains the different ANN using different subsets. Then it determines membership grades of these subsets and combines them via a new ANN to get final results. The KDD CUP 1999 is used for the experimental. It incorporates both the training phase and testing phase. The results show that FC-ANN gets higher accuracy superior to naïve Bayes and BPNN.

Another approach to network anomaly detection, based on dynamic self-organizing maps (DSOM) and ant colony optimization (ACO) clustering is proposed by [16]. The proposed work has four stages: the DSOM which is an unsupervised neural network used to determine the shapes as well as the size of network during the training. ACO clustering is used to choose the objects and cluster them from the output layer of DSOM according to the shortest distance. The labelling clusters algorithm is the third stage which is labelled the objects as the 'normal' cluster, or labelled as the 'anomalous' cluster based on DSOM and ACO clustering. The last stage is the detection algorithm which depends on Bayes theorem. Experiments on the KDD99 dataset showed higher performance than SVM and K-NN. However, the results reported were not numerical which make it very hard to compare with likely systems. Similar works done by the authors can also be found at [17] and [18].

2.3 Bees Algorithm

2.3.1 Real Bees

Bees live in high-organized societies that communicate together and exchange information about the food sources using some action movement called a "waggle dance". The waggle dance contains specific signals that reveal important information. For example, if the bee wants to advertise about new location for a food source, a series of waggle dance runs with semicircular round pointed to direction related to the sun's position and x-axis. The longer duration of dancing means the more profitable source which is as a consequent will attract the watcher bees to visit the place. For more details about the foraging process, reader is referred to [8].

2.3.2 Bees Algorithm (BA)

BA divides the bees into groups based on there labor. The scout bees(n), elite bees(e), and recruited bees for selected sites (nep) and recruited bees for non-selected sites($m-e$). Also the sites are divided into elite sites (which are visited by elite bees)(e), other sites which are not selected by elite bees(m). Beside that, there are other factors required by BA to be set, the initial size of each patch (ngh) (a patch is a region in the search space that includes the visited site and its neighbourhood), and the stopping criterion. The pseudo code of BA is shown in Figure 2.

1. Initialise population with random solutions.
2. Evaluate fitness of the population.
3. While (stopping criterion not met) //Forming new population.
4. Select sites for neighbourhood search.
5. Recruit bees for selected sites (more bees for the best e sites) and evaluate fitnesses.
6. Select the fittest bee from each site.
7. Assign remaining bees to search randomly and evaluate their fitnesses.
8. End While.

Fig. 2. Pseudo Code of BA

BA generates n scout randomly distributed initial population of the search space. After initialization, the fitness of the sites visited by the scout bees is evaluated in Step 2. The bees that have the highest fitness are selected as the "elite bees" and sites visited by them are chosen for neighbourhood search in Step 4. In Step 5, the bees are recruited for the selected sites and more employed bees for the elite sites (which are visited by the elite bees). At the same step, the fitnesses are evaluated. The differential recruitments beside the scouting are very significant operation of the BA. Then based on the evaluation fitness in previous step, the fittest bee is chosen to be selected to

generate the next bee population in Step 6. This addition is required to reduce the number of points to be explored, however, it is not utilized in nature. In Step 7, the remaining bees in the population are distributed randomly to scout for new solutions around the search space.

BA conducts searches the solution that minimizes the given cost measure by locating in the most promising solutions, and explores their neighbourhoods by looking for the global minimum of the objective function. Distribution of publications with respect to years is also given in Table 1. From Table 1 it is clear that the interest of researchers and the applications of the algorithms proposed on BA raises day by day and the number of papers in the literature increases exponentially.

Table 1. Recent Publications of BA

Citation	Problem Study /Application
[8]	Continuous Optimization
[9]	LVQ-Neural Network
[10]	Optimization of neural networks for wood defect
[19]	Manufacturing cell formation
[20]	Applications of the BA in engineering design
[21]	Data clustering
[22]	Using the bees algorithm to schedule jobs
[24]	Protein conformational search
[25]	Synthesizing multiple beam antenna arrays
[23]	Optimal design of mechanical components
[26]	Interference suppression of linear antenna arrays

BA also was used instead of a back propagation algorithm to optimize the weights of the neural network. As [9] presented an application of the BA to optimize neural networks for the identification of defects in wood veneer sheets. BA was successfully used to train the LVQ and the MLP neural networks for control chart pattern recognition. Despite the high dimensionality of these problems, the BA succeeded in training more accurate classifiers than those produced by the standard LVQ training algorithm and the backpropagation algorithm.

The training phase of the proposed HoneybeeGuard is different in such a way that we use different learning methods, different approach, different classifier system, and different data. In particular, the choice is to use classifiers that apply the proposed HoneybeeGuard approach, RBF learning method, data mining, and for intrusion detection process. To the best of our knowledge, the construction of training neural network by the BA to improve the performance of intrusion detection system has not been addressed in the literature. In this paper, the BA extends to classification and demonstrates its effectiveness in intrusion detection.

3 System Architecture and Design

A HoneybeeGuard approach is proposed to be applied in IDS. The approach consists of three functional computational components. These components are:

1. Undesirable-Absent (UA),
2. Desirable-Present (DP), and
3. Filtering-Decision (FD).

The hierarchical hybrid strategy for the proposed approach is illustrated in Figure3. The figure demonstrates a streamlined, centralized intrusion detection design that consists of both the misuse detection method and the anomaly detection method. The workloads are distributed to multiple detectors to monitor the intrusion detection process efficiently.

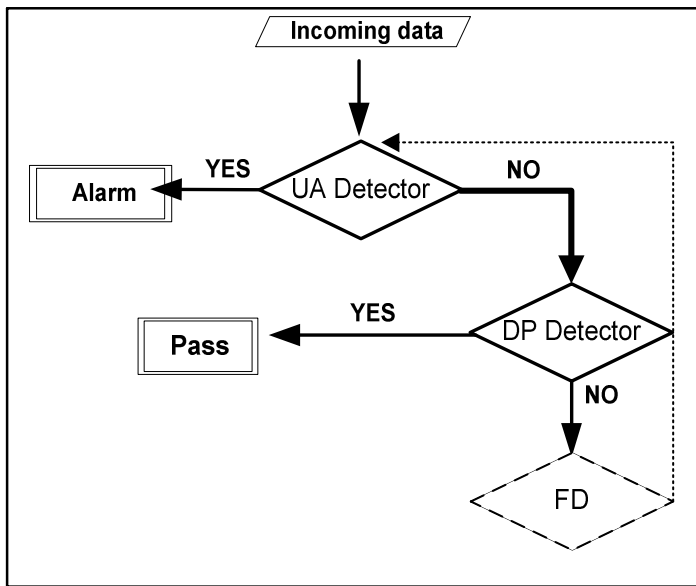


Fig. 3. The Proposed HoneybeeGuard Approach

In nature, honey bee guard used either the UA or the DP in filtering the incomer, not both. The honeybee guard would accept the incomers because they do not contain the undesirable characteristics *U-absent*; these undesirable characteristics are seen on almost all intruders but not on valid incomers. On the other hand, the honeybee guard would also accept the incomers because they have the desirable characteristics, *D-present*. These characteristics would be seen on most entries.

However, in IDS field it may not be practical to use one of the methods alone without combining it to the other because of the two reasons:

- If a system uses only the **D-present**, that will make a large number of acceptance errors and smaller amount of rejection errors. Figure 4(a) exhibits the consequences of using only the D-present on determining the acceptance and rejection as **permissive guard**.
- If a system uses only the **U-absent**, there will be more rejection errors and a smaller amount of acceptance errors. It would be stricter to admit intruders but may also be more likely to reject valid incomers. Figure 4(b) indicates the consequences of using only the U-absent on determining the acceptance and rejection as **restrictive guard**.

In the proposed HoneybeeGuard approach, we combine between both UA and DP while there is no such combining in nature. This is necessary here to reduce the number of errors in acceptance and rejection and to get the full advantages from them all. Figure 4(c) shows the consequences of combining between the UA and DP on determining the acceptance and rejection as an **optimal guard**.

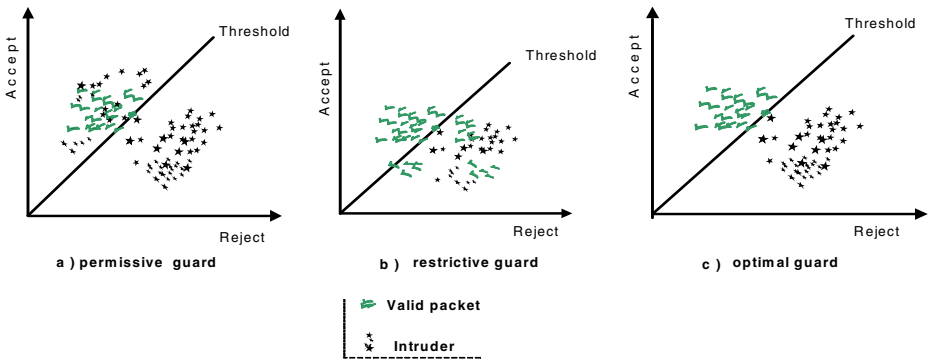


Fig. 4. Resultant of using different threshold (a) the permissive

3.1 Undesirable-Absent (UA) Method

Every receiving record will be verified and examined at UA method [13]. In UA detector the innocuous records will be allowed to pass whereas the intrusion ones will be detected. In order to apply the idea of UA in the domain of intrusion detection, we need to determine the characteristics that will represent the malicious or attacks (the non-nestmate in nature).

The dataset collected by DARAPA and preprocessed for the KDD '99 competition have several relevant features that can be used as characteristics for the attack properties. The information of the attack is included in each packet to aid in classification processes.

Figure 5 illustrates the UA training and testing phases. Neural network receives data from the data set and analyzes it for misuse intrusion. There are several advantages to this approach. It has the ability to learn the characteristics of misuse attacks and identify instances that are unlike any which have been observed during the training. It has a high degree of accuracy to recognize known suspicious events.

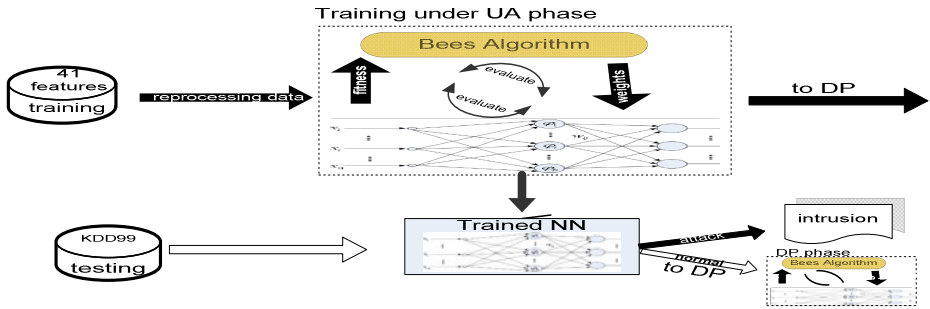


Fig. 5. The UA Training and Testing Overview

3.2 Desirable-Present (DP) Method

Forwarding records from *UA* are checked here for two reasons: 1) we assume that there will be some malicious attacks which couldn't detected by the *UA* yet. 2) this hierarchy structure supports our goal to detect attacks with both misuse and anomaly techniques efficiently [13]. *DP* compares between characteristics of the forwarded packet and its "template" which contains the desirable characteristics of an accepted packet. The abnormal packets will be forwarded to *FD*.

DP detector is built of normal data. The purely normal of KDD '99 dataset which is free from attacks is used to train the *DP* detector in order to recognize the desirable characteristics of the incoming connection records. The advantage of that is to assist *DP* to detect new types of intrusions; as the new intrusions will deviate from normal network. The most important requirement during the training is to train *DP* detector by purely normal data. If data contains some attacks within the training data, detector may not detect such of these attacks on the future as these attacks are assumed that they are normal

In addition to that, normal traffic features are extracted and considered as desirable characteristics. Each feature category provides information that can be used to discriminate between normal and abnormal traffic. After preprocessing the data and training the neural network, the task of *DP* is to determine whether test data belongs to normal or to abnormal. The result of learning process is a neural network which able to detect anomalies in the traffic during the testing phase.

In "10% KDD" there are 97.277 records contain normal traffic. These records "free from attack" are used to construct the *DP*. During the test phase, *DP* uses a classifier which has been previously trained to classify the suspicious connections that forwarded from *UA* as anomalous or abnormal type of records. On other words, the normal packets will pass through the classifier but the intrusion packets will be detected and forwarded into *FD* classifier.

3.3 Filtering-Decision (FD)

In natural honeybee behavior, the earlier studies indicated the ability of honeybees to develop templates by referring to groups of individuals [11]. The information about the intruders is renewed from time to time [12]. The natural honeybee studies show

the importance of the colony odour template of guard bees as the main factor in changing acceptance. A changeable template would be adaptively valuable by increasing the discriminator's accuracy [5].

In this context, the Filtering-Decision (*FD*) is taking the part of the template updated for *UA*. In *FD* method, the packets which have been detected as abnormal and forwarded from *DP* are stored and verified by *FD*. When more attacks are detected and stored, a clustering algorithm *K-Means* is used to cluster these stored attacks into groups based on their types. The motivation of using *K-Means* algorithm [27] to group the stored data is because of the simplicity, the ease of implementation and the high performance of the *K-Means*. After these groups reaching a predetermined threshold, the *FD* will make these intrusions records flow back to train the *UA* detector.

Clustering process in *FD* is very simple and it is used to minimize the efforts and time that may spent by the administrator to classify the incoming records. However, the *FD* decision is not final; it is stored as suspended in a quarantine status waiting for the user evaluation to see the label manually and forward the matched group to *UA* detector. The framework of *FD* is shown below in Fig.7.

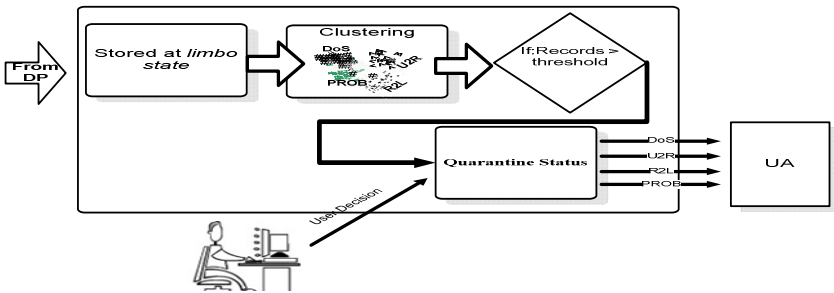


Fig. 7. The *FD* Framework Overview

Before doing the experiments we need to setup the neural network and *BA* parameters. Regarding the neural network configurations, it is based on the processing of the methods. As it is in *UA*, neural network involves three layers: an input layer, a hidden layer, and an output layer. The input layer has 41 neurons, one for each point in a pattern. The output layer comprises 5 neurons, one for each of the five classes. *DP* has a different task which is to distinguish between normal and abnormal. Therefore the architecture of neural network is configured differently in some parts to fulfil this task. On another word, as the number of output differs between the two methods two different configurations of neural network were used. The output is either normal or anomalous and the output layer comprises 2 neurons. The rest of neural network configuration is almost same as *UA*.

Regarding the *BA* configuration, the following parameters were determined according to preliminary experiments: population size *n* of 300; number of selected points *m* 10; number of elite points out of *m* selected points *e* 2; initial patch size *ng* 0.1; number bees for elite points *nep* 80; number of bees for other selected points *nsp* 20.

4 Evaluation Criteria

- **Accuracy or Detection rate:** A system that has 80% accuracy is a system that correctly classifies 80 instances out of 100 in their actual class.
- **False Positive and Negative:** A positive data is considered to be an attack data, while a negative data is considered to be a normal data.
- **Confusion Matrix:** Displays the number of correct and incorrect predictions made by the model compared with the actual classifications in the test data [28].

4.1 Experimental Result

4.1.1 UA Results

In this experiment, four types of attack data (PROBE, DoS, R2L, U2R) and normal data were used to test the Performance of UA detector. Moreover, most challenges to UA are to identify known and unknown intrusions and classify them to their four major classes as most intrusion detection systems fail to overcome this task [28]. The result for UA detector is shown in Table 4.

Table 4. The Performance of the UA Detector

	DR (%)	FPR (%)
Dos	99.30	0.77
R2L	89.03	1.70
Probe	98.21	2.17
U2R	93.16	0.90

From Table 4 we can deduce the efficient task of the UA detector. DoS got the highest detection rate (99.30%) and the lowest false alarm rate (0.30%). Probe is the next higher DR, got (98.21%). This can be explained by the fact that NN network learned more about DoS and Probe during learning process because of the majority presentation of theirs records during learning data set (10% KDD). R2L and U2R attack categories also got a high detection rate (89.03%, 93.16% respectively) but not high as DoS and Probe, this can be explained because of the lack of their presentations during the learning phase.

Table 5 below also provides a numerical measure about the performance more than other matrices. The Confusion matrices used in the table 5 provided more details and can't be biased since they measure what percentage of intrusions the system is able to detect and how many incorrect classifications are made in the process. First row in Table 5, for example, 626 *DoS* instances were classified as *Normal*, whilst 22824 *DoS* instances were classified as *DoS*, etc.

Table 5. Confusion matrix for the UA

	NORMAL	DoS	PROBE	U2R	R2L
DoS	626	228244	983	0	0
PROBE	0	75	4091	0	0
U2R	0	0	0	212	16
R2L	0	62	0	1714	14413

From the previous experiments it can be seen that most intrusion connections have been filtered out by UA. The number of total instances found by UA detector is **246960**, which considered as **98.6%** of test dataset. Among these instances, only 3476 records are wrongly classified. The result shows that the UA detector highly correctly learns the patterns of various attacks and is excellent in classifying most types of attacks. Remain records of the test dataset after UA detector are **64069** records which contain normal records and the rest of attacks, known and unknown, not detected by UA. Nevertheless, the DP detector will receive the flow of records and filter out the intrusion connections based on desirable characteristics.

4.1.2 DP Detector Experiments

The result of DP testing is shown in Table 6. It shows the power of the trained neural network in identifying the unknown intrusion by detecting the deviation of normal. DP has the overall detection rate of 97.30% and 2.30% false positive rate. We can notice from Table 6 that DP detects more anomalies and intrusion than UA but with a little bit increasing of false alarm. Hence, there is a trade off here. More restrict condition for a connection to be normal will result more false alarms. According to [29], anomaly detectors perform better than other detectors over KDD’99 dataset using various machine learning algorithms but with higher false alarm. One explanation to this might be due to the complex distribution of training samples and embedded attack patterns in the KDD’99 data.

Table 6. The Performance of the DP Detector

	Correctly Classified	False Classified	Detection Rate(%)	FPR (%)
Normal	62339	1730	97.30	2.30

The last step is to measure the overall performance. The results show that the system gets high DR (more than 99.4%) and low FP (less than 0.5%). When we compare these results to the results before updating UA detector, it is notable that the DP detector has its advantages and able automatically and adaptively to detect new unknown attacks, which were not included into the training set.

4.3 Comparison with Related Approaches

Table 7 provides summary of results from recent approaches related to the proposed study. The performance of the proposed approach has been compared with some other machine learning methods as shown below.

Table 7. Results from Related Approaches

Approach	Detection Rate (%)	FP Rate (%)
The Proposed Honeybee Approach	99.4	0.5
FC-ANN Approach [15]	96.71	NA
PSOSVM [4]	96.11	3.89
ESC-IDS [29]	95.3	1.9
Winner of KDD [30]	91.8	0.6
Hybrid NN [31]	90	5
Neuro-Fuzzy [32]	NA	5.07

From the Table 7, it is clear that the results show that the proposed IDS is better at detecting attacks than other related works. Furthermore, a false positive rate for the proposed honeybee approach is lower compared to others approaches.

5 Conclusion

The focus of this research was to demonstrate how productive the crossover between biology and computer science can be. The detection system in honeybee which keeps the colony safe was the basis frame for the main approach of the research to improve the effectiveness of IDS. Characterizing the incoming packets to support detection was significant. Characterization methods have ranged using neural network trained by BA that it becomes perceptive and sensitized to detect intrusions.

Within UA phase, the focusing was on the attack features. Neural network was trained to recognize the characteristics of attacks in order to classify these characteristics as undesirable characteristics during the testing phase. The advantages were the ability to learn the characteristics of misuse attacks and identify instances that didn't observe during the training. Moreover, UA has a high degree of accuracy to recognize known suspicious events.

The strength of DP was to its ability to recognize novel attacks. DP detector was built of normal data in order to recognize the desirable characteristics of the incoming connection records. The advantage was to support DP to detect new types of intrusions which deviated from normal network and forwarded to FD.

The FD method was taking part of the template updated for UA. The packets which are forwarded from DP and have been detected as abnormal are stored and verified here. This procedure made UA more effective by updating its classifier with new records (novel attacks) in real-time. The idea of FD was to update the UA structure automatically and adaptively according to novel intrusions identified by a clustering program.

To examine the feasibility of our approach, we conducted several experiments and comparisons. The proposed approach was compared with related approaches using the DARPA KDD'99 benchmark dataset. In addition, we selected other datasets and subsets to evaluate the proposed approach in different domains. The experimental results demonstrate that the proposed approach can improve the detection deficiency issue by reducing the false alerts and increasing the detection accuracy.

As a final word, the work done in this paper introduced a new scenario for the IDS deployment. The paper is about more than IDS. The honeybee approach is applicable to several other domains related to security issues; such as: designing security policy, extending to IPS..etc.

Acknowledgment

The authors would like to thank the members in the Computer Security and Forensic Lab and also Security Research Group for their helpful discussions and suggestions. This work was supported by RU Grant No.1001/PKOMP/817048 and No.1001/PKOMP/822126, School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia.

References

- [1] CSO, Deloitte's Center for Security & Privacy Solutions (2010), <http://www.csoonline.com>
- [2] Anderson, J. P.: Computer security threat monitoring and surveillance. Technical report, James P. Anderson Co., Fort Washington, Pennsylvania (April 1980)
- [3] Rains, G.C., Tomberlin, J.K., Kulasiri, D.: Using insect sniffing devices for detection. *Trends in Biotechnology* 26(6), 288–294 (2008)
- [4] Srinoy, S.: Intrusion Detection Model Based On Particle Swarm Optimization and Support Vector Machine. In: *Computational Intelligence in Security and Defense Applications*, CISDA 2007, pp. 186–192. IEEE Computer Society Press, Los Alamitos (2007)
- [5] Couvillon, M.J., et al.: En garde: rapid shifts in honeybee, *Apis mellifera*, guarding behaviour are triggered by onslaught of conspecific intruders. *Animal Behaviour* 76(5), 1653–1658 (2008)
- [6] Butler, C.G., The, F.J.: behaviour of worker honeybees at the hive entrance. *Behaviour* 4, 263–291 (1952)
- [7] Stein, G., Chen, B., Wu, A.S., Hua, K.A.: Decision tree classifier for network intrusion detection with GA-based feature selection. In: *Proceedings of the 43rd annual Southeast regional conference - Volume 2 (ACM-SE 43)*, vol. 2, pp. 136–141. ACM, New York (2005), doi:10.1145/1167253.1167288
- [8] Pham, D.T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M.: The bees algorithm—a novel tool for complex optimisation problems. In: *Proceedings of IPROMS*, Conference, Cardiff, UK, pp. 454–461 (2006a)
- [9] Pham, D.T., Ghanbarzadeh, A., Koc, E., Otri, S.: Application of the bees algorithm to the training of radial basis function networks for control chart pattern recognition. In: *Proceedings of 5th CIRP international seminar on intelligent computation in manufacturing engineering (CIRP ICME 2006)*, Ischia, Italy (2006b)

- [10] Pham, D.T., Koc, E., Ghanbarzadeh, A., Otri, S.: Optimisation of the weights of multi-layered perceptrons using the bees algorithm. In: Proceedings of 5th international symposium on intelligent manufacturing systems (2006)
- [11] Kitching, I.J.: Phylogeny of the death's head hawkmoths, *Acherontia*[*Laspeyres*], and related genera (Lepidoptera: Sphingidae: Sphinginae: Acherontiini. *Systematic Entomology*, 71–88 (2003), doi:10.1046/j.1365-3113
- [12] Breed, D.E., Guzmán-Novoa, G.J.: 3 Hunt, Defensive behavior of honey bees: organization, genetics, and comparisons with other Bees. *Annual Review of Entomology* 49, 271–298 (2004)
- [13] Ali, G.A., Jantan, A., Ali, A.: Honeybee-Based Model to Detect Intrusion. In: Park, J.H., Chen, H.-H., Atiquzzaman, M., Lee, C., Kim, T.-h., Yeo, S.-S. (eds.) *ISA 2009. LNCS*, vol. 5576, pp. 598–607. Springer, Heidelberg (2009)
- [14] Ryan, J., Lin, M.J., Mikkulainen, R.: *Intrusion detection with neural networks*. MIT Press, Cambridge (1998)
- [15] Wang, G., Hao, J., Ma, J., Huang, L.: A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering. *Expert Syst. Appl.* 37(9), 102 (2010), doi:10.1016/j.eswa.2010.02.102
- [16] Feng, Y., Zhong, J., Xiong, Z., Ye, C., Wu, K.: Network Anomaly Detection Based on DSOM and ACO Clustering. In: Liu, D., Fei, S., Hou, Z., Zhang, H., Sun, C. (eds.) *ISNN 2007. LNCS*, vol. 4492, pp. 947–955. Springer, Heidelberg (2007), http://dx.doi.org/10.1007/978-3-540-72393-6_113
- [17] Feng, Y.Z., Wu, K., Wu, Z.: An unsupervised anomaly intrusion detection algorithm based on swarm intelligence. In: Feng, Y.Z., Wu, K., Wu, Z. (eds.) *Proceedings of 2005 International Conference on Machine Learning and Cybernetics*, vol. 7, pp. 3965–3969. IEEE Computer Society Press, Los Alamitos (2005)
- [18] Feng, Y.J., Zhong, J., Ye, C., Wu, Z.: Clustering based on self-organizing ant colony networks with application to intrusion detection. In: Ceballos, S. (ed.) *Proceedings of 6th International Conference on Intelligent Systems Design and Applications (ISDA 2006)*, Jinan, China, pp. 3871–3875. IEEE Computer Society Press, Washington, DC, USA (2006)
- [19] Pham, D.T., Afify, A., Koc, E.: Manufacturing cell formation using the bees algorithm. In: *IPROMS 2007: Innovative Production Machines and Systems Virtual Conference*, Cardiff, UK (2007)
- [20] Pham, D.T., Castellani, M., Ghanbarzadeh, A.: Preliminary design using the bees algorithm. In: *Proceedings of Eighth International Conference on Laser Metrology, CMM and Machine tool Performance, LAMDAMAP*, Euspen, Cardiff, UK, pp. 420–429 (2007)
- [21] Pham, D.T., Otri, S., Afify, A.A., Mahmuddin, M., Al-Jabbouli, H.: Data clustering using the bees algorithm. In: *Proceedings of 40th CIRP International Manufacturing Systems Seminar* (2007)
- [22] Pham, D.T., Koc, E., Lee, J., Phruksanant, J.: Using the bees algorithm to schedule jobs for a machine. In: *Proceedings of Eighth International Conference on Lasermetrology, CMM and machine tool performance*, pp. 430–439 (2007h)
- [23] Pham, D.T., Ghanbarzadeh, A., Otri, S., Koç, E.: Optimal design of mechanical components using the Bees Algorithm. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* (May 1, 2009), doi:10.1243/09544062JMES838
- [24] Bahamish, H., Abdullah, R., Salam, R.: Protein conformational search using bees algorithm. In: *AICMS 2008: Second Asia International Conference on Modeling and Simulation*, pp. 911–916 (2008)

- [25] Guney, K., Onay, M.: Bees algorithm for design of dual-beam linear antenna arrays with digital attenuators and digital phase shifters. *Int. J. RF Microw Comput-Aided Eng.* 18(4), 337–347 (2008)
- [26] Guney, K., Onay, M.: Bees algorithm for interference suppression of linear antenna arrays by controlling the phase-only and both the amplitude and phase. *Expert Systems with Applications* 37(4), 957–4174 (2010), doi:10.1016/j.eswa.2009.09.072, ISSN 0957-4174
- [27] Spath, H.: *Clustering Analysis Algorithms*, p. 1980. Wiley, Chichester
- [28] Ali, G.A., Lu, W., Tavallaee, M.: *Network Intrusion Detection and Prevention: Concepts and Techniques*, 1st edn. Springer Publishing Company, Heidelberg (2009) (Incorporated)
- [29] Toosi, A.N., Kahani, M.: A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers. *Computer communications* 30, 2201–2212 (2007)
- [30] Pfahringer: Winning the KDD99 classification cup: Bagged boosting. *KDD 1999* 1(2), 67–75 (2000)
- [31] Jirapummin, C., Wattanapongsakorn, N., Kanthamanon, P.: Hybrid neural networks for intrusion detection systems. In: *Proc. of The 2002 International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2002)*, pp. 928–931 (2002)
- [32] Alshammari, R., Sonamthiang, S., Teimouri, M., Riordan, D.: Using neuro-fuzzy approach to reduce false positive alerts. In: *Fifth Annual Conference on Communication Networks and Services Research (CNSR 2007)*, pp. 345–349. IEEE Computer Society Press, Los Alamitos (2007)

Multi-classifier Scheme with Low-Level Visual Feature for Adult Image Classification

Mohammadmehdi Bozorgi, Mohd Aizaini Maarof, and Lee Zhi Sam

Faculty of Computer Science & Information Systems,
Universiti Teknologi Malaysia,
81310 UTM Skudai, Johor, Malaysia
bmohammadmehdi2@live.utm.my, aizaini@utm.my,
samleecomp@gmail.com

Abstract. As the usage and accessing of children to the web resources with porn images contain is growing, requirement of methods with high accuracy to detect and block adult images is a necessity. In this paper, a novel multi-classifier scheme is proposed based on low-level feature to exploit of advantages in classifier ensemble for achieving better accuracy compared to single classifier that applied to adult images detection. Low-level features are three different MPEG-7 descriptors include Color Layout Descriptor (CLD), Scalable Color Descriptor (SCD) and Edge Histogram Descriptor (EHD). In the classification part Support Vector Machine (SVM) and AdaBoost are applied and combined. Experimental results indicate that proposed scheme works better than each single classifier that used in the experiments.

Keywords: Adult Image, Classification, Visual Features, MPEG-7 Descriptor, Classifier ensemble.

1 Introduction

Image classification is one of the most important parts of image analysis. As the large number of images for on-line accessing is growing very fast, indexing of images became necessitate for the last decades; To overcome on considered crisis, numerous image classification methods have been developed and applied in several diverse areas and image categories.

Nowadays using of Internet and popularity of WebPages become more and more especially among children and on the other hand numbers of unsuitable websites with the porn image contents are quickly growing. It is a very necessary for having a good filtering method to detect adult images and prevent to access of this kind of images [1].

Recent approaches in adult image detection methods show many classification techniques is used by researchers to achieve high accurate method [2] [3] [4] [5]. Using of different classifiers such as Neural Network (NN) classifier, Support Vector Machine (SVM), AdaBoost and also fuzzy classifier in adult image domain proves that only one specific classifier are not able to produce faultless result.

Multi-classifier is one of solution that has been used for that kind of problems which does not have a specified way to solve [6] [7]. In many fields combination of classifiers it used especially in pattern recognition techniques like person recognition, remote sensing and medical applications [7]. In this paper, getting advantage of previous successful applied classifiers for adult image detection and also using of advantages from Multi-classifier, new classifier is proposed.

The rest of the paper is organized as follows: Section 2 is the brief explanation of the previous and related works; section 3 outlines the proposed adult image classification scheme. Experimental results are presented in Section 4. Finally, conclusion is drawn in Section 5.

2 Related Works

In the past decade, researchers attempt to proposed and design a method to detect adult images among any other type of images. The first approach to detect naked people has implemented by [8].

Previous efforts for adult image detection methods include two main elements: 1) Image Feature and 2) Image Classification. Feature based methods emphasize the drawing out features in pornographic images [9]. These extracted features can detect adult images by self or can be used as the input for one specific image classification. Features in images also can classify in two categories: low level and high level visual feature.

Low level visual features provide basic and necessary information about image [10]. For example [11] used skin color model at YIQ and YUV color space and, edge and texture information and also color histogram as low level visual features to perform adult image detection.

However, low level visual feature is recognized to be adequate to get reliable accuracy [12], but using of only low level visual features has some weaknesses. For example when the given image is a bikinied women the level of different between various features are very low among bikinied women and naked women [3]. Face is one of high level visual feature that have been used in adult image detection [13].

Many image classification achievements have been proposed and commonly used for the adult image classification. Neural network (NN) is one of the classifiers that used in many classifier problems. [2] and [14] have used NN with Mpeg-7 descriptors as the input. Neural networks needs high rate of data set to provide high accurate result. Hence Support Vector Machine (SVM) that has similar origin as neural network is used for image classification with limited training dataset and also SVM shows very good performance in image classification in general. [3] have applied SVM in different way of implementation for adult image detection.

AdaBoost is another classifier that has been used for adult image detection. The speed of training and convergence are very depending to the implementation. [13] found that AdaBoost can be very comparative choose for the adult image classification.

In the all of the previous works and because of lack of standard datasets in the case of study there is no fair comparison. Obviously there is no faultless method and different methods can show this fact that multi-classifier scheme is one solution in the case of adult image detection.

3 Proposed Scheme

In the proposed scheme as many other adult image detection it start by extracting features from given images and after a simple preprocessing which is almost normalization of extracted features, data pass to the classifier for the making decision. Figure 1 illustrates overall scheme flow.

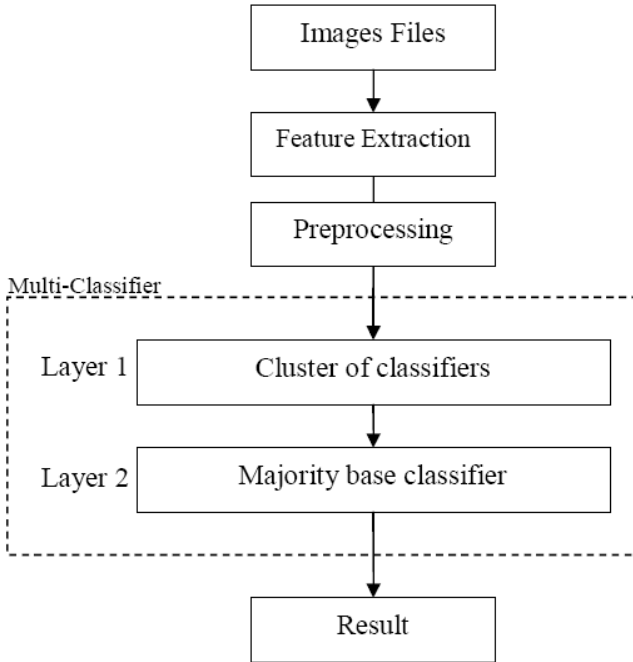


Fig. 1. Overall scheme follow

3.1 Image Files

One of the reasons of using Color Layout Descriptor (CLD), Structure Color Descriptor (SCD) and Edge Histogram Descriptor (EHD). from MPEG7 descriptors is that all of these descriptors are not depend on any image format or image size. These descriptors can apply on all images without agonize for matching image size and image formats. Consequently, any images can be selected for the input to be classified.

3.2 Feature Extraction

Collecting information from images and sending to the classifier is the feature extraction that can be done directly (low-level) or indirectly (semantic). Feature extraction is the first step to achieve the goal so selection of features can be very

important for accomplishing enhanced outcome. Without having dominant features, classifiers are not able to perform classification in accurate state. For the feature extraction in this study MPEG-7 descriptor [16] is used as low-level feature.

Both low-level and semantic visual features have own advantages. In low-level visual features, complexity of getting information is extremely low and reliability of given information are complete. However, using of semantic visual features like skin color or breast shape would affect to the result depending of the skin color detector accuracy.

MPEG-7 standard indicate a group of descriptors to supply standardized descriptions of audio and video content. Each descriptor explains low-level visual feature for example color and shape [17]. In this study to classify adult image three MPEG-7 visual descriptors are used which are Color Layout Descriptor (CLD) [18], Scalable Color Descriptor (SCD) [19] and Edge Histogram Descriptor (EHD) [15] that is shown in figure 2.

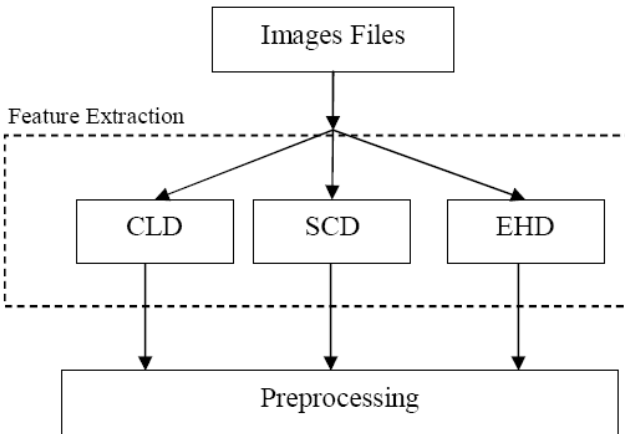


Fig. 2. Feature Extraction

Table 1. Three MPEG-7 descriptors description

Descriptor Name	Number of features	Min-Max Value (after quantization)	Description
CLD	18	0-255	Coefficients of DCT transform from Y, Cb and Cr
SCD	64	0-15	Extracts a quantized HSV color histogram
EHD	80	0-7	Distribution of non-directional edges and non-edge cases as well as four directional edges

The structure of each descriptor is important to justify how the classifiers in the next steps should be implemented. Table 1 describes that CLD is the coefficients of

DCT transform of the images and SCD is the color histogram in HSV color space of the image and finally, EHD is the edge information of the image that shows in a histogram format.

3.2.1 Color Layout Descriptor (CLD)

Color Layout Descriptor, in a very compressed structure shows the color division. The functionality of this descriptor can be high accurate visual matching without high fiscal rate. This descriptor has two advantages as stated in bellow:

1. This descriptor can use in all images or frames of videos even with different resolutions. It means this descriptor works with any format, bit depths and resolutions.
2. This descriptor use of very small hardware and software resources to produce the results. In both withdrawal and similarity checking process only eight (8) bytes need for a single picture or frame.

3.2.2 Scalable Color Descriptor (SCD)

Scalable Color Descriptor can capture not only color content but also can catch structure information for any color content in an image. First task of this descriptor is like a simple color histogram. But different to the color histogram, where the structures of places in two images are dissimilar to each other but with same capacity of colors, the descriptor does not give same answer and result. Color space which use in this descriptor is HMMD. Color values are quantized none uniformly into 256, 128, 64 or 32 bins. SCD in contrast to normal color histogram give more performance, especially for the natural images.

3.2.3 Edge Histogram Descriptor (EHD)

Five kinds of edge distribution are symbolized by EHD. It make of 4 directional edges, and also one none directional edge. While edges are important parts of image view, usage of this descriptor is very high. By combination of EHD and Color Histogram Descriptor (CHD), image retrieval methods can have sufficient efficiency.

3.3 Preprocessing

In this section all extracted features will normalize to the numbers between 0 and 1. Performing this normalization will help to SVM and AdaBoost to produce more accurate result. for the reason that the extracted features are quantized before normalizing, process of normalization is included only division by the maximum value in each feature(255 for CLD, 15 for SCD and 7 for EHD).

3.4 Multi-classifier

The proposed classification method is designed with two layers. First layer is more vital and second layer produce final result. In the first layer according to multi classification method a group of classifiers instead of using a single classifier is employed.

To detect which classifiers should be included in the first layer of adult image classifier, literature review has some guide line. In the previous and related works SVM and also AdaBoost have been developed and got more popularity among other

classifiers and each of them has advantages and disadvantages. But popularity and also efficiency of these two classifiers which are discussed in literature review are the reasons that in this proposed method, these two classifiers are used.

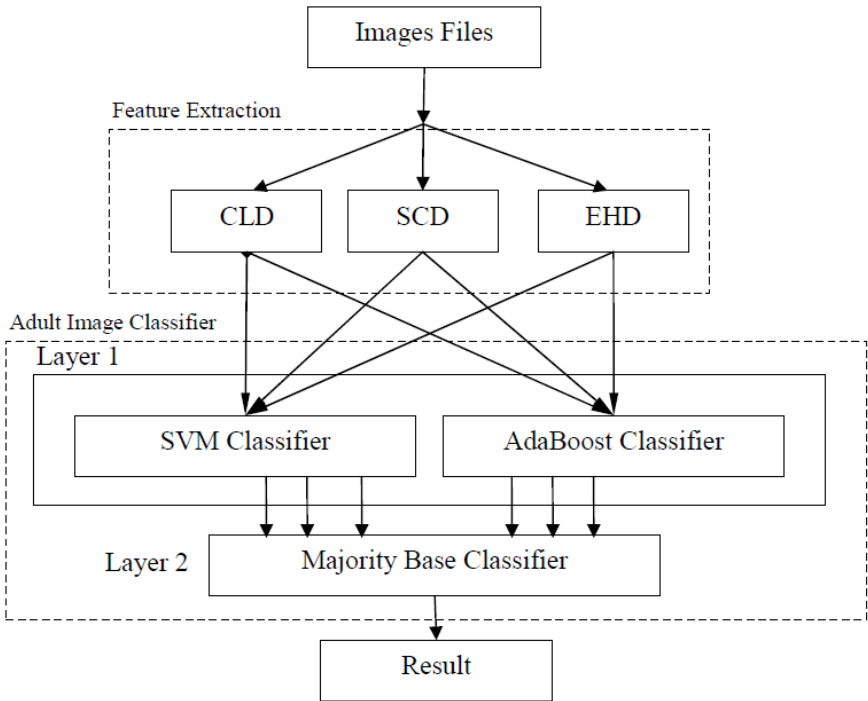


Fig. 3. Proposed adult image classifier scheme

As it shows in figure 3, the adult image classifier is proposed, and each of these classifiers making the own result for each of descriptors on a single image and finally passed it to the next layer.

At the next layer, “Majority Based Classifier (MBC)” with a single and straightforward calculation makes the final result. At equation (1) the calculation of MBC is showed. In (1), n is referring to the number of classifiers in the first layer and also $\delta(i)$ is the result of classifier number i that can be in one of these two forms: 1 (for adult images) and 0 (for non-adult images).

$$\begin{aligned}
 & \text{if } \left(\sum_{i=1}^n \delta(i) > \frac{n}{2} \right) \Rightarrow \text{adult - image} \\
 & \text{else} \Rightarrow \text{non - adult - image}
 \end{aligned}
 \tag{1}$$

4 Simulation and Result

4.1 Dataset

From the literatures it can be detected that there is no standard dataset in this specific case of research. Other researchers used of own collected dataset.

For our case we collect 1000 of adult images that mostly collected from pinkworld.com and also 2000 of non-adult images are selected from imageafter.com for training classifiers. In addition 322 adult images and 317 non-adult images are used for testing the proposed system.

4.2 Experimental Result

For the first step of proving proposed multi-classifier scheme, each single classifier tested separately for each single MPEG-7 Descriptor. As stated in the table 2, SVM on the EHD works best by around 80% in True-Positive(TP) and also same classifier and same descriptor by near 4% got best result for False-Positive(FP) in single classifier testing.

Table 2. Single classifier result for each MPEG-7 Descriptor

	CLD		SCD		EHD	
	TP	FP	TP	FP	TP	FP
SVM	76.4%	12.3%	64.6%	6.6%	79.8%	4.1%
AdaBoost	78.3%	14.8%	70.5%	11.7%	73.3%	5.7%

Before continue and showing the result for making better understanding we gave each classifier-input one number and it is exposed in the table 3.

Table 3. Numbering of Classifiers

	CLD	SCD	EHD
SVM	1	2	3
AdaBoost	4	5	6

To indicate that which combination of classifiers is best, 26 possibilities are tested and demonstrated in table 4. The second layer of multi-classifier is majority based classifier and it limit the inputs to the odd number of inputs for passing to the second layer. After testing the result was like table 4.

Table 4. All combinations and results in proposed Multi-Classifer scheme

Combination	TP	FP	Combination	TP	FP
1+2+3	88.8	3.5	2+4+5	75.5	7.2
1+2+4	89.1	8.5	2+4+6	84.2	3.1
1+2+5	74.8	7.2	2+5+6	74.2	5.4
1+2+6	84.6	3.1	3+4+5	88.5	5
1+3+4	91.6	8.2	3+4+6	91.9	4.4
1+3+5	89.1	5.4	3+5+6	85.7	4.4
1+3+6	89.1	4.7	4+5+6	85.4	5
1+4+5	88.5	9.8	1+2+3+4+5	88.5	5
1+4+6	90.7	8.5	1+2+3+4+6	90.4	2.8
1+5+6	85.1	5.7	1+2+3+5+6	85.4	2.5
2+3+4	88.8	3.2	1+2+4+5+6	84.8	5
2+3+5	75.8	4.7	1+3+4+5+6	90.1	4.4
2+3+6	83.5	3.8	2+3+4+5+6	85.4	2.8

In terms of TP, combinations of classifiers number 3, 4 and 6 got the best rank among other groups by 91.9% accuracy and also for the best FP combination of 5 classifiers which are 1, 2, 3, 5 and 6 reach 2.5% of mistakes. According to the table 3 for the numbering of classifiers we can determine best combination for FP is to hybrid SVM and AdaBoost classifier with usage of EHD feature as input and also AdaBoost with CLD feature. Similarly, SVM with CLD, SCD and EHD plus AdaBoost with SCD and EHD can achieve best FP.

4.3 Evaluation of Results

For the evaluating of proposed classification scheme, both FP and TP was considered so differentiate between FP and TP make the final decision for evaluating and ranking each classifier.

Table 5. Performance Ranking

Rank	Combination	TP-FP	Rank	Combination	TP-FP	Rank	Combination	TP-FP
1	1+2+3+4+6	87.6	8	3+4+5	83.5	15	3+5+6	81.3
2	3+4+6	87.5	9	1+2+3+4+5	83.5	16	2+4+6	81.1
3	1+3+4+5+6	85.7	10	1+3+4	83.4	17	1+2+4	80.6
4	2+3+4	85.6	11	1+2+3+5+6	82.9	18	4+5+6	80.4
5	1+2+3	85.3	12	2+3+4+5+6	82.6	19	1+2+4+5+6	79.8
6	1+3+6	84.4	13	1+4+6	82.2	20	2+3+6	79.7
7	1+3+5	83.7	14	1+2+6	81.5	21	1+5+6	79.4

For each single classifier SVM with input of EHD got the best result with 75.5(79.8-4.1=75.5). With applying this pattern to the proposed scheme the ranking table 5 illustrated.

Combination of 1, 2, 3, 4 and 6 and also 3, 4 and 6 got the best result among the other combinations and the difference between TP and FP was around 87.5 which was more than 12% for the both TP and FP more than the best single classifier(SVM over EHD input). To compare this result with average of single classifiers that used in 3, 4, 6 which was 67% $((TP3-FP3) + (TP4-FP4) + (TP6-FP6))/3$ the 20.5% the accuracy had improvement.

5 Conclusion

Obviously, results from proposed scheme prove better accuracy compare to single usage of classifiers. The combination of different classifiers and also with different features gives this opportunity to the Multi-Classifier scheme to the make better decision. Between three different classifier sometimes it happened that 2 of them give correct answer and one of them make a mistake, it help to the multi-classifier to cover the mistake from each other.

On the other hand always more classifier dose not have a meaning to have better accuracy as it stated in the table 5, between the first 8 best combinations only 2 of them used of 5 classifier and the rest use only 3 classifier.

For the future work we can add more classifiers and also more feature to get better and faultless output.

References

1. Vijayendar, G., SakthiPriyaBalaji, R.: Integrated Approach to Block Adult Images in Websites. In: International Conference on Computer Technology and Development icctd, vol. 2, pp. 421–425 (2009)
2. Kim, W., Lee, H.-K., Yoo, S.-J., Baik, S.W.: Neural network based adult image classification. In: Duch, W., Kacprzyk, J., Oja, E., Zadrozny, S. (eds.) ICANN 2005. LNCS, vol. 3696, pp. 481–486. Springer, Heidelberg (2005)
3. Jeon, J.H., Kim, S.M., Choi, J.Y., Min, H.S., Ro, Y.M.: Semantic Detection of Adult Image Using Semantic Features. In: 4th International Conference on Multimedia and Ubiquitous Engineering (MUE), pp. 1–4 (2010)
4. Choi, B., Chung, B., Ryou, J.: Adult Image Detection Using Bayesian Decision Rule Weighted by SVM Probability. In: Fourth International Conference on Computer Sciences and Convergence Information Technology, ICCIT 2009, pp. 659–662 (2009)
5. Liu, B.B., Su, J.Y., Lu, Z.M., Li, Z.: Pornographic Images Detection Based on CBIR and Skin Analysis. In: Fourth International Conference on Semantics, Knowledge and Grid, pp. 487–488 (2008)
6. Kuncheva, L.I.: Combining Pattern Classifiers: Methods and Algorithms (2004)
7. Oza, N.K., Tumer, K.: Classifier ensembles: Select real-world applications. *Journal of Information Fusion* 9, 4–20 (2008)
8. Fleck, M., Forsyth, D., Bregler, C.: Finding Naked People. In: European Conference on Computer Vision, vol. II, pp. 592–602 (1996)

9. Wang, X., Hu, C., Yao, S.: A Breast Detecting Algorithm for Adult Image Recognition. In: International Conference on Information Management, Innovation Management and Industrial Engineering, vol. 4, pp. 341–344 (2009)
10. Agbinya, J., Lok, B., Sze Wong, Y., Silva, S.: Automatic online porn detection and tracking. In: 12th IEEE Int. Conf. on Telecommunications, pp. 1–5 (2005)
11. Jiao, F., Gao, W., Duan, L., Cui, G.: Detecting adult image using multiple features. In: International Conferences on Info-tech and Info-net, ICII 2001, vol. 3, pp. 378–383 (2001)
12. Lee, J.-S., Kuo, Y.M., Chung, P.C., Chen, E.L.: Naked image detection based on adaptive and extensible skin color model. *Pattern Recognition* 40(8), 2261–2270 (2007)
13. Zheng, Q.F., Zeng, W., Wen, G., Wang, W.Q.: Shape-based adult images detection. In: Third International Conference on Image and Graphics, pp. 150–153 (2004)
14. Kim, W., Lee, H.-K., Park, J., Yoon, K.: Multi class adult image classification using neural networks. In: Kégl, B., Lee, H.-H. (eds.) Canadian AI 2005. LNCS (LNAI), vol. 3501, pp. 222–226. Springer, Heidelberg (2005)
15. Eom, M., Y Choe, Y.: Fast Extraction of Edge Histogram in DCT Domain based on MPEG7. *Proceedings of World Academy of Science, Engineering and Technology* 9, 209–212 (2005)
16. Manjunath, Salembier, P., Sikora, T.: Introduction to MPEG-7: Multimedia Content Description Interface. Wiley & Sons, Chichester (2002) ISBN 0-471-48678-7
17. Furht, B.: MPEG-7 Applications. In: *Encyclopedia of Multimedia*, pp. 473–475. Springer, US (2008)
18. Kasutani, E., Yamada, A.: The MPEG-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval. In: International Conference on Image Processing, vol. 1, pp. 674–677 (2001)
19. Manjunath, B.S., Ohm, J.-R., Vasudevan, V.V., Yamada, A.: Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology* 11(6), 703–715 (2001)

Managing Communications Challenges in Requirement Elicitation

Noraini Che Pa¹ and Abdullah Mohd Zin²

¹ Department of Information System
Faculty of Computer Science and Information System
University Putra Malaysia
norainip@fsktm.upm.edu.my

² Programming and Software Technology Research Group
Faculty of Information Science and Technology
University Kebangsaan Malaysia
amz@ftsm.ukm.my

Abstract. Eliciting requirements for a system is an important activity in requirement engineering. This process involves communication between customers and developers. Researchers have identified that poor communication is one of the most common problems in identifying and defining customer's requirements. Managing communication is challenging and difficult as this process also includes cognitive aspect, personalities, techniques and tools. An study was carried out to explore the communication challenges that are faced by developers and customers during software requirement elicitation in Malaysia. The results of this study had shown that there are some important communication challenges during software requirements elicitation process in Malaysia. In addition, most of the practitioners were pursuing the practices that have been traditionally used by the organizations. Consequently, the results of this study have given us a good incentive to expand our research in the area of software requirements elicitation.

Keywords: Requirement elicitation, software requirements, communication.

1 Introduction

Requirements elicitation is a process of searching, revealing, acquiring and detailing of requirements for computer based system [1]. The effects of poor software requirements include cost rework, budget overruns, poor quality systems, stakeholders' dissatisfaction and projects failure. In order to solve this problem, a number of researches have been carried out to study the elicitation requirements practices, problems and tools. One of the important issues in requirement elicitation is the communication between customers and developers, which include the cognitive aspect, personalities, techniques and tools [2,3]. The issue of communication skills and analyst-client relationship has been a consistent issue in IS literature for over 20 years [4]. Communication problem has also been identified as one of major factors that caused delay and failure of software projects [5].

The purpose of this paper is to identify some of communication challenges that may arise during requirement elicitation and propose some possible interventions that may be carried out in order to mitigate these challenges. This paper is divided into five parts. The second section of the paper describes the communication model in general. In the third section, we describe the communication activities and processes in requirements elicitation. The fourth section discusses the research that we have carried out to identify some communication challenges in the process of software development in Malaysia. The last part is the conclusion.

2 Communication Model

There are a lot of communication model which have been proposed by researchers from several disciplines. The communication elements most frequently mentioned are source-receiver, encoder-decoder, feedback, message, noise, context and effect [6]. The first model discussed is linear by Shannon–Weaver. This model describes the communication process as having information sources, a message, a transmitter, a signal, a receiver, a destination and noise. Transactional model is another communication model described by Boone [7]. This model involves two or more participants who act and react to one another. A message can be successfully exchanged only when both the sender and the receiver perceive it in the same way. This process relies on feedback from the receiver to the sender, and is influenced by both the context in which the communication process occurs, and the channel chosen for the transmission of that message. Boone also noted that the perception of the receiver is critical in effective communication [7]. Another model stated that the meaning of a message does not reside completely in the message, but is constituted by the receiver based on their own background [8]. Due to differences in background, this meaning can differ considerably from the intended meaning of the sender.

Communication is more than just an expression, but actually requires participants to share qualities such as language, experience, cultural values and knowledge [6]. It has been agreed that communication occurs within a particular context, and this context has at least four dimensions: physical, social, psychological and temporal. The physical dimension refers to the physical environment in which the communication occurs and may exert some influence on the content as well as the form of the message. The social dimension reflects the relationships between the participants and the norms and cultures of the society in which they are communicating. The psychological context consists of such aspects as the friendliness or unfriendliness and the formality or informality. The temporal dimension includes the time at which the communication takes place. Berlo's model emphasizes that communication is an interactive process, without beginning, end, or a fixed order of events [8]. He specifies that four significant elements of communication. are source, message, channel and receiver. The channel for sending and receiving messages consists of five human senses: seeing, hearing, touching, smelling and tasting. A perception and personality screen for each participant in the communication process is highlighted by [10]. The argument by Kerzner is that a message will be encoded with factors of the sender's personality and perceptions of the environment, the context, the message and the receiver, and their self-perceptions. Besides that, the receiver of

the messages, were influenced by their own perceptions and personality. Barnlund model, explains that cues are signals that a person processes from the environment [9]. Another model illustrated by Wenburg represents the communication process as an infinity symbol. This model demonstrates that communication is a never-ending process. This model can also be expanded by similar loops to indicate several participants in the communication transaction [9].

Based on the various models of communication that has been described above, we can simplify that a model of communication consists of six elements as shown in Figure 1. This model includes sources that encode the message, the channel or medium through which the messages are transmitted, noise that interfere with the communication process, a receiver who decode it, and feedback that is sent to the source.

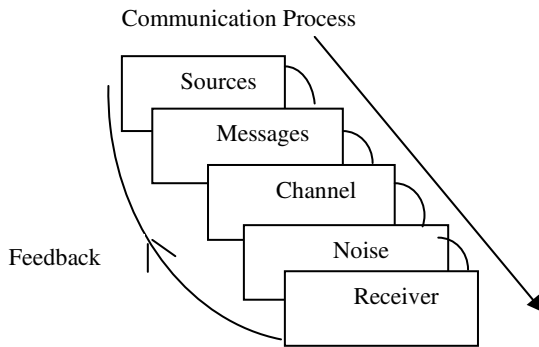


Fig. 1. A simple communication model

3 Communication Process and Activity in Requirements Elicitation

Communication activities in requirement elicitation could be categorized into knowledge elicitation, negotiation and integration [11][12].

1. Knowledge elicitation: This process includes sharing an understanding on ideology, vision, knowledge, experience and technology.
2. Knowledge negotiation: This process includes negotiating for software requirements information.
3. Knowledge integration: This process involves the accepting of strategy and software requirements.

We can use the model described in Figure 1 as a model of communication in a requirement elicitation process as described in Figure 2. The source is the customer, the message is the requirements, the channel is the technique, the noise is the communication challenges, the receiver is the developer and the feedback is the software requirements specification. In other words, the connections between communication theory and requirements elicitation go beyond the simplistic of the communication fallacy.

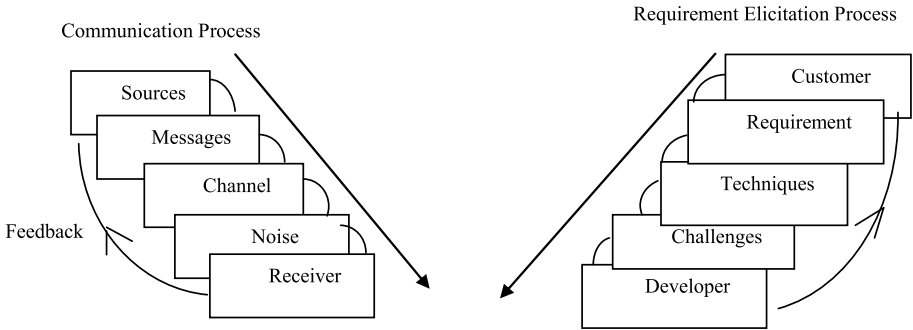


Fig. 2. Communication and Requirements Elicitation Process

Coughlan et al [13] had developed a communication framework that can be used for requirement elicitation. This framework consists of selection and participation of stakeholders, stakeholder’s interaction, communication activities and use of communication techniques. This framework also considers other aspects such as cultural factor, political, communication plan, approach and purpose of interaction. Several possible challenges that may arise during communication activities have also been identified (as indicated in Table 1). These challenges need to be reduced in order to ensure effective communication.

Table 1. Challenges in Communications Activities

Activities	Problems
Knowledge Elicitation	<ol style="list-style-type: none"> 1. Gap in understanding 2. Thinking in innovative 3. Overlap in aspect
Knowledge Negotiation	<ol style="list-style-type: none"> 1. Commitment 2. Mutual perspective
Knowledge Acceptance	<ol style="list-style-type: none"> 3. Information exchange 1. Feedback 2. Fear factor 3. Change management

Source: Coughlan et al. (2003)

4 Managing Communication during Requirement Elicitation in Malaysia

The general objective of the study is to investigate the processes and issues of managing communication during requirements elicitation activities between customers and developers.

4.1 Research Approach

This study was carried out by using a questionnaire survey and case study approaches. The questionnaire includes questions on managing communication and the challenges to this activity. The results of the survey have been analyzed using SPSS. This approach is suitable to gather broad information of the study.

Table 2 shows respondent distribution according to sector. They are from various agencies that are categorized as government agencies, semi government, private agencies MSC status and non MSC status. Analysis of data shows that 42.9% respondents are from the private agencies Multimedia Super Corridor (MSC) status, 33.3% from private agencies non-MSC status, 21.4% from government agencies and 2.4% from semi government.

Table 2. Respondent Distribution according to Sector

Sector	Frequency	Percentage (%)
Government agencies	9	21.4
Semi-government	1	2.4
Private agencies (MSC status)	18	42.9
Private agencies (MSC non-status)	14	33.3
Total	42	100

Table 3 shows respondent distribution according to respective positions. They are individual who involved in activity along requirement elicitation process. Analysis of data shows that most respondents are project leader 52.4%, analyst 21.4%, software engineer 2.4%, programming 4.8% and others 19.0%.

Table 3. Respondent Distribution according to Position

Position	Frequency	Percentage (%)
Project Leader	22	52.4
Software engineer	1	2.4
System analyst	9	21.4
Programmer	2	4.8
Others	8	19.0
Total	42	100

Requirements sources are information that was gathered from the customers. These refer to customer needs for the new or upgrading system implementation. From the analysis, it is shown that numerous sources were used in process identification requirements. These sources come from customers. Respondent chose work process as their main source to identify software requirements. Other sources used are based from existing system, organization rules, expert knowledge, document and others source.

Many organizations choose and modify their sources in accordance with technology changes. Besides that, sources of project are also influenced by changes of other factors such as economic, politic, social, regulations, financial, psychology, history and geography. For example an organization that practices a biroracy system can cause difficulty in gathering requirements comparing to others. Moreover, the changes of management and political pattern in an organization also influence in delivering the requirements sources. These new changes made some customers feel unhappy and unable to accept. Rarely, changes in requirements and scope will affect on changes of information delivered. Also information that was prepared becomes inconsistent. Information was delivered through email, telephone and interview. Information which received by email is easier to understand compared to other medium.

To know requirements elicitation practice that implemented, a few issues relate to selection technique and factor which influenced that selection and process have been stated to respondents. The respondents were requested to state in the questionnaire one or more techniques that were used for requirements elicitation process. These techniques that were used by respondent were listed in Table 4.

Table 4. Requirements Elicitation Techniques

Eliciting Techniques	Frequency	Percentage(%)
Interview	34	81
Survey	15	35.7
Scenario	12	28.6
Document Analysis	25	59.5
Questionnaire	13	31
Focus Group	9	21.4
Workshop	8	19
Use Case	8	19
Requirement Reuse	2	4.8

Table 4 shows requirements elicitation techniques that chosen by respondent. The analysis shows 34 from 42 respondents (81%) chosen interview as technique that most suitable for software requirement elicitation. While 15 from 42 respondents (35.7%) choose survey technique, 25 (59.5%) choose document study technique, 13 (31%) choose questionnaire, 12 (28.6%) choose scenario, 9 (21.4%) choose focus group, 8 (19%) choose workshop, 8 (19%) choose use case and 2 (4.8%) choose Requirement Reuse technique that most not chosen by respondent. This retrieval is pursuant past research which found interview technique is technique that most popularly used for software requirements elicitation process.

4.2 Challenges

In order to study the problem of communication between customers and developers with more detail, a case study has been carried out. The study involved nine projects. From

the analysis, (refer Table 5) the results of the study show that communication problems can be divided into five topics namely type of input, personalities involved, communication skills, medium of communication and procedures. The problems occurred in the delivery of input probably because the information is ambiguous, requirements and scopes are frequently changed. Besides that, a lot of information presented is outside of the customer’s field of expertise. Therefore the customers neglect focusing on delivering the information. As a result, the developer feels that the information delivered by the customers is ambiguous and not consistent. This is because the customers do not understand their roles in system development. In addition, the medium and procedures also contributed to the challenges in managing communication.

Table 5. Challenges in Managing Communication

Criteria	Challenges
Type of input	<ol style="list-style-type: none"> 1. ambiguity and not clear of information 2. redundancy of information 3. frequent requirements changes 4. different information 5. changes of scope
Personalities involved	<ol style="list-style-type: none"> 1. changes of staff 2. lack of cooperation 3. lack of comittment and participation 4. less tolerance 5. lack control of work burden 6. lack of ability to handle conflicts
Communication skills	<ol style="list-style-type: none"> 1. lack of ability in solving the ambiguity 2. lack of ability in proactive and instructive information delivery 3. lack of communication skills (verbal) 4. lack of presentation skill 5. lack of logic written 6. lack of organizing an idea 7. lack of sorting information
Medium of communication	<ol style="list-style-type: none"> 1. late of responses 2. interpretation mistake 3. cannot access file 4. information not consistent 5. informal information 6. unrecorded information
Procedures	<ol style="list-style-type: none"> 1. changes of report frequently 2. changes of report types 3. changes of document 4. changes of management and political rules 5. changes of criteria acceptance

Sometimes, requirements and scopes are frequently changed and this affected the information that is delivered. Often the information was delivered using different methods, thus problems occur since individual may have different views and understanding of a given topic.

4.3 Proposed Intervention for Managing the Challenges

Some intervention needs to be carried out in order to mitigate the effect due to communication challenges that occur between customers and developers. The lists of intervention suggested are given in Table 6.

One of the challenges faced by developers is to understand the customer's real requirement. This is a difficult task since most of the customers do not understand computer and system terminologies. Thus it is possible that when they mention certain terms, they are actually referring to different things. To mitigate this problem, it is important that customers should have some basic knowledge about computers and systems.

Since most of the communication between customers and developers are in written form (emails, letters, documents etc), it is important that customers must be able to express the requirements without any ambiguities.

The third challenge is the medium. Currently most of the communication between customers and developers are done either through face-to-face oral communication (interview, meeting etc), letters or memos, and emails. There is a need to improve the medium of communication in order to reduce the possibility of misunderstanding.

The fourth and fifth challenges are related to developers' knowledge and ability to express the requirements properly.

Table 6. Intervention Steps for Managing Communication Challenges

Challenges	Intervention Steps
Customer Knowledge	Provide knowledge to customer so that they can describe the problems better.
Customer Expression	Provide checklist guide to customer for requirements that input completely and understand by developer.
Medium	Provide communication facilities to customer to allow customer to communicate for problem solving that occur during communication process such as lately and misunderstanding.
Developer Knowledge	Provide facilities to developer for writing requirements specification document and checking written content.
Developer Expression	Provide facilities to generate document that written by developer into specification document type that can be modify by developer

5 Conclusion

This paper discusses managing communication challenges among customer and developer during requirements elicitation process. The knowledge in the intervention steps can be used by the customer to express their requirements. Hence it can reduce from getting incorrect input such as the ambiguous information and frequently changed requirements and scopes. Intervention steps also provide the communication facilities for the customer to discuss any information regarding the requirements. These intervention steps will be used in the process model to develop system in assisting communication between customer and developer during requirements elicitation. It is agreed that effective communication is not easy to achieve, but this intervention steps can be used to assist in managing communication challenges. By completing and adequately manage the communication challenges, the good requirements can be created. Requirements document is always taken as the basis for software development.

References

1. Coulin, C., Sahraoui, El Kader, A., Zowghi, D.: Towards a Collaborative and Combination Approach to Requirements Elicitation within a Systems Engineering Framework. In: 18th International Conference on System Engineering, pp. 456–461 (2005)
2. Coughlan, J., Mark, L., Robert, D.M.: Communication issues in requirements elicitation: a content analysis of stakeholder experiences. *Information and Software Technology* 45, 525–537 (2003)
3. Aurum, A., Wohlin, C.: *Engineering And Managing Software Requirements* (2005)
4. Urquhart, C.: Analysts and Clients in Conversation: Cases in Early Requirements Gathering. In: *Proceeding. of the International Conference on Information Systems*, pp. 115–127 (1998)
5. Curtis, B., Krasner, H., Iscoe, N.: A Field study of the software design process for large system. *Communications of the ACM* 31(11), 1268–1286 (1988)
6. Devito, J.A.: *Communicology: An Introduction To The Study Of Communication*. Harper & Row, New York (1982)
7. Boone, L.E., Kurtz, D.L.: *Contemporary Business Communication*. Prentice Hall Inc, Englewood Cliffs (1994)
8. Bruckmann, C., Hartley, P.: *Business Communication*. Routledge, London (2002)
9. Lewis, P.V.: *Organizational Communication: The Essence Effective management*, 2nd edn. Grid Columbus Ohio Inc. (1980)
10. Kerzner, H.: *Project Management: a system approach to planning, scheduling and controlling*. Van Nostrand Reinhold Company Inc., NewYork (1995)
11. Walz, D., Elam, J., Curtis, B.: Inside a software design team: Knowledge acquisition, Sharing and Integration. *Communications of the ACM* 36(10), 62–77 (1993)
12. Ocker, R., Fjermestad, J., Hiltz, S.R., Turoff, M.: An exploratory Comparison of Four Modes of Communication for Determining Requirements: Results on Creativity, Quality and satisfaction. In: *Proc. 13 Hawaii International*, pp. 568–577 (1997)
13. Coughlan, J., Macredie, M.: Effective Communication in Requirements Elicitation: A Comparison of Methodologies. *Requirements Engineering* 7(2), 47–60 (2002)

Learning Efficiency Improvement of Back Propagation Algorithm by Adaptively Changing Gain Parameter together with Momentum and Learning Rate

Norhamreeza Abdul Hamid, Nazri Mohd Nawi, Rozaida Ghazali,
and Mohd Najib Mohd Salleh

Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia,
86400 Parit Raja, Batu Pahat, Johor, Malaysia
gi090007@siswa.uthm.edu.my, {nazri,rozaida,najib}@uthm.edu.my

Abstract. In some practical NN applications, fast response to external events within enormously short time is highly demanded. However, by using back propagation (BP) based on gradient descent optimization method obviously not satisfy in several application due to serious problems associated with BP which are slow learning convergence velocity and confinement to shallow minima. Over the years, many improvements and modifications of the back propagation learning algorithm have been reported. In this research, we modified existing back propagation learning algorithm with adaptive gain by adaptively change the momentum coefficient and learning rate. In learning the patterns, the simulation results indicate that the proposed algorithm can hasten up the convergence behaviour as well as slide the network through shallow local minima compare to conventional BP algorithm. We use three common benchmark classification problems to illustrate the improvement of the proposed algorithm.

Keywords: back propagation, convergence speed, shallow minima, adaptive gain, adaptive momentum, adaptive learning rate.

1 Introduction

Multilayer Feedforward Neural Network (MLFNN) also referred to as Multilayer Perceptron (MLP) is one of the most popular and most frequently used type of Neural Network (NN) models due to its clear architecture and comparably simple algorithm. It can unravel classification problems implicating non-linearly separable patterns and can be used as a comprehensive function generator [1]. Due to its ability to solve some problems with relative ease of use, robustness to noisily input data, execution speed and analyzing complicated systems without accurate modelling in advance, MLFNN has successfully been implemented across an extraordinary range of problem domains that involves prediction and a wide ranging usage area in the classification problems [2-9].

The MLP is composed by a set of sensorial units organized in three hierarchical of layers comprise of the input layer of neurons, one or more intermediary or hidden layer of neurons and the output layer of neurons. The consecutive layers are fully connected. The connections between the neurons of adjacent layers relay the output signals from one layer to the next. Throughout the learning phase, the interconnections are optimized to minimize the predefined function.

Among the existing paradigms, Back Propagation (BP) algorithm is a supervised learning procedure for training MLFNN which is based on the gradient descent (GD) optimization method that endeavors to minimize the error of the network by moving down the gradient of the error curve [1]. This algorithm mapping the input values to the desired output through the network. This output pattern (actual output) is then compared to the desired output and the error signal is computed for each output unit. The signals are then transmit backward from the output layer to each unit in the transitional layer that contributes directly to the output and the weights are adjusted iteratively during the learning process.

In some practical NN applications, fast response to external events within tremendously short time are highly demanded and expected. However, the comprehensively used of BP algorithm based on GD optimization method obviously not satisfy in many applications especially large scale application and when higher learning accuracy as well as generalization performances are obligatory. The reason for this unsatisfaction is due to the slow learning convergence velocity though the network has achieved stopping criteria. Moreover, it also frequently confinement to shallow minima.

It is noted that many local minima complications are closely associated to the neuron saturation in the hidden layer. When such saturation exists, neuron in the hidden layer will lose their sensitivity to the input signals and propagation chain is blocked severely. In some situation, the network can no longer learn. Furthermore, the convergence behaviour of the BP algorithm also depends on the selection of network architecture, initial weights and biases, learning rate, momentum coefficient, activation function and value of the gain in the activation function.

In the recent years with the progress of researches and applications, the NN technology has been enhanced and sophisticated. Research has been done on modification of the conventional BP algorithm in order to improve the efficiency and performance of MLP network training. Much work has been devoted to improve the generalization ability of the networks. These implicated the development of heuristic techniques, based on properties studies of the conventional back propagation algorithm. These techniques include such idea as varying the learning rate, using momentum and gain tuning of activation function. Lera *et al.* [10] described the use of Levenberg-Marquardt algorithm for training multi-layer feed forward neural networks. Though, the training times required strongly depend on neighbourhood size. Meanwhile, Wang *et al.* [11] proposed an improved BP algorithm caused by neuron saturation in the hidden layer. Each training pattern has its own activation function of hidden nodes in order to prevent neuron saturation when the network output has not acquired the desired signals. The activation functions are adjusted by the adaptation of gain parameters during the learning process. However, this approach not performed well on the large problems and practical applications. Otair and Salameh [12] designed the optical back propagation (OBP) algorithm which is applied

on the output units. This kind of algorithm used for training process that depends on a multilayer NN with a very small learning rate, especially when using a large training set size. Conversely, it does not guarantee to converge at global minima because if the error closes to maximum, the OBP error grows increasingly. While Ji *et al.* [13] proposed a back propagation algorithm that improved conjugate gradient (CG) based. In the CG algorithm, a search is performed along conjugate directions which usually lead to faster convergence compared to gradient descent directions. Nevertheless, if it reaches a local minimum, it remains forever, as there is no mechanism for this algorithm to escape.

Nazri *et al.* [14] demonstrated that by adaptively change the ‘gain’ value for each node can significantly reduce the training time without modifying the network topology. Therefore, this research proposed a further improvement on [14] by adjusting activation function of neurons in the hidden layer in each training patterns. The activation functions are adjusted by the adaptation of gain parameters together with adaptive momentum and adaptive learning rate value during the learning process. The proposed algorithm, back propagation gradient descent with adaptive gain, adaptive momentum and adaptive learning rate (BPGD-AGAMAL) significantly can obviate the network from trapping into shallow minima that caused by the neuron saturation in the hidden layer as well as hasten up the convergence behaviour. In order to verify the efficiency of the proposed algorithm, the performance of the proposed algorithm will be compared with the conventional BP algorithm and back propagation gradient descent with adaptive gain (BPGD-AG) proposed by [14]. Some simulation experiments were performed on three classification problems including glass [15] soybean [16] and breast cancer Wisconsin [17].

The remaining of the paper is organized as follows. In Section 2, the effect of using activation function with adaptive gain is reviewed. While in Section 3 presents the proposed algorithm. The performance of the proposed algorithm is simulated on benchmark dataset problems in Section 4. This paper is concluded in the final section.

2 The Effect of Using Activation Function with Adaptive Gain Parameter together with Momentum Coefficient and Learning Rate on the Performance of Back Propagation Algorithm

An activation function is used for limiting the amplitude of the output neuron. It generates an output value for a node in a predefined range as the closed unit interval $[0,1]$ or alternatively $[-1,1]$ which can be a linear or non-linear function. This value is a function of the weighted inputs of the corresponding node. The most commonly used activation function is the logistic sigmoid activation function. Alternative choices are the hyperbolic tangent, linear, step activation functions. For the j^{th} node, a logistic sigmoid activation function which has a range of $[0,1]$ is a function of the following variables, viz:

$$o_j = \frac{1}{1 + e^{-c \cdot a_{net,j}}} \quad (1)$$

where,

$$a_{net,j} = \left(\sum_{i=1}^l w_{ij} o_i \right) + \theta_j \quad (2)$$

where,

- o_j output of the j^{th} unit.
- o_i output of the i^{th} unit.
- w_{ij} weight of the link from unit i to unit j .
- $a_{net,j}$ net input activation function for the j^{th} unit.
- θ_j bias for the j^{th} unit.
- c_j gain of the activation function.

The value of the gain parameter, c_j , directly influences the slope of the activation function. For large gain values ($c \geq 1$), the activation function approaches a ‘step function’ whereas for small gain values ($0 < c \leq 1$), the output values change from zero to unity over a large range of the weighted sum of the input values and the sigmoid function approximates a linear function.

Most of the application oriented papers on NN tend to advocates that NN operate like a ‘magic black box’, which can simulate the “learning from example” ability of our brain with the help of network parameters such as weights, biases, gain, hidden nodes, and so forth. Also, a unit value for gain has generally being used for most of the research reported in the literature, though a few authors have researched the relationship of the gain parameter with other parameters which used in back-propagation algorithms.

The learning rate (LR) is one of the most effective means to accelerate the convergence of BP learning. It is a crucial factor to control the variable of the neuron weight adjustments at each iteration during the training process and therefore affects the convergence rate. In fact, the convergence speed is highly depending on the choice of LR. The LR values need to be set appropriately since it dominate the performance of the BP algorithm. The algorithm will take longer time to converge or may never converge if the LR is too small. On the contrary, the network will accelerate the convergence rate significantly and still possibly will cause the instability whereas the algorithm may oscillate on the ideal path if the LR value is too high. The value of LR usually set to be constant, which means that the selected value is employed for all weights in the whole learning process. Yet, Ye [18] stated that the constant learning rate of the BP algorithm fails to optimize the search for the optimal weight combination. Hence, a search methodology has been classified as a “blind-search”.

Another effective approach regarding to hasten up the convergence and stabilise the training procedure is by adding some momentum coefficient (MC) to the network. Moreover, with MC, the network can slide through shallow local minima. Formerly, the MC is typically preferred to be constant in the interval $[0,1]$. In spite of that, it is

discovered from simulations that the fixed momentum coefficient value seems to hasten up learning only when the recent downhill gradient of the error function and the last change in weight have a parallel direction. When the recent negative gradient is in a crossing direction to the previous update, the MC may cause the weight to be altered up the slope of the error surface as opposed to down the slope as preferred. This leads to the emergence of diverse schemes for adjusting the MC value adaptively instead of being kept constant throughout the training process.

Results in [19] demonstrate that the LR, MC and gain of the activation function have a significant impact on training speed. Thimm *et al.* [20] also proved that a relationship between the gain value, a set of initial weight values, and LR value exists. Eom *et al.* [21] proposed a method for automatic gain tuning using a fuzzy logic system. Nazri *et al.* [14] proposed a method to change the gain value adaptively on other optimisation method such as CG. Hamid *et al.* [22] demonstrated that adaptive momentum coefficient and adaptive gain of the activation function significantly improved the training time.

3 The Proposed Algorithm

In this section, a further improvement on the current working algorithm proposed by [14] for improving the training efficiency of back propagation is proposed. The proposed algorithm modifies the initial search direction by changing the three terms adaptively for each node. Those three terms are; gain value, MC and LR. The advantages of using an adaptive gain value together with MC and LR have been explored. Gain update expressions as well as weight and bias update expressions for output and hidden nodes have also been proposed. These expressions have been derived using same principles as used in deriving weight updating expressions.

The following iterative algorithm is proposed for the batch mode of training. The weights, biases, gains, LRs and MCs are calculated and updated for the entire training set which is being presented to the network.

```

For a given epoch,
  For each input vector,
    Step 1.
    Calculate the weight and bias values using the previously
    converged gain, MC and LR value.
    Step 2.
    Use the weight and bias value calculated in Step (1) to
    calculate the new gain, MC and LR value.
    Repeat Steps (1) and (2) for each input vector and sum all the
    weights, biases, LR, MC and gain updating terms
    Update the weights, biases, gains, MCs and LRs using the summed
    updating terms and repeat this procedure on epoch-by-epoch basis
    until the error on the entire training data set reduces to a
    predefined value.
  
```

The gain update expression for a gradient descent method is calculated by differentiating the following error term E with respect to the corresponding gain parameter. The network error E is defined as follows:

$$E = \frac{1}{2} \sum (t_k - o_k(o_j, c_k))^2 \quad (3)$$

For output unit, $\frac{\partial E}{\partial c_k}$ needs to be calculated while for hidden units, $\frac{\partial E}{\partial c_j}$ is also required.

The respective gain values would then be updated with the following equations:

$$\Delta c_k = \eta \left(-\frac{\partial E}{\partial c_k} \right) \quad (4)$$

$$\Delta c_j = \eta \left(-\frac{\partial E}{\partial c_j} \right) \quad (5)$$

$$\frac{\partial E}{\partial c_k} = -(t_k - o_k) o_k (1 - o_k) \left(\sum w_{jk} o_j + \theta_k \right) \quad (6)$$

Therefore, the gain update expression for links connecting to output nodes is:

$$\Delta c_k(n+1) = \eta (t_k - o_k) o_k (1 - o_k) \left(\sum w_{jk} o_j + \theta_k \right) \quad (7)$$

$$\frac{\partial E}{\partial c_j} = \left[-\sum_k c_k w_{jk} o_k (1 - o_k) (t_k - o_k) \right] o_j (1 - o_j) \left(\left(\sum_j w_{ij} o_i \right) + \theta_j \right) \quad (8)$$

and the gain update expression for the links connecting hidden nodes is:

$$\Delta c_j(n+1) = \eta \left[-\sum_k c_k w_{jk} o_k (1 - o_k) (t_k - o_k) \right] o_j (1 - o_j) \left(\left(\sum_j w_{ij} o_i \right) + \theta_j \right) \quad (9)$$

Similarly, the weight and bias expressions are calculated as follows:

The weight updates expression for the links connecting to output nodes:

$$\Delta w_{jk} = \eta (t_k - o_k) o_k (1 - o_k) c_k o_j \quad (10)$$

Similarly, the bias update expressions for the output nodes would be:

$$\Delta \theta_k = \eta (t_k - o_k) o_k (1 - o_k) c_k \quad (11)$$

The weight update expression for the links connecting to hidden nodes is:

$$\Delta w_{ij} = \eta \left[\sum_k c_k w_{jk} o_k (1 - o_k) (t_k - o_k) \right] c_j o_j (1 - o_j) o_i \quad (12)$$

Similarly, the bias update expressions for the hidden nodes would be:

$$\Delta\theta_j = \eta \left[\sum_k c_k w_{jk} o_k (1 - o_k) (t_k - o_k) \right] c_j o_j (1 - o_j) \quad (13)$$

4 Results and Discussions

The performance criterion used in this research focuses on the speed of convergence, measured in number of iterations and CPU time as well as accuracy. The real world classification problem datasets are obtained from UCI Machine Learning Repository at Centre for Machine Learning and Intelligent Systems have been used to verify our algorithm. Three classification have been tested including glass [15], soybean [16] and breast cancer Wisconsin [17].

The simulations have been carried out on a Pentium IV with 2 GHz HP Workstation, 3.25 GB RAM and using MATLAB version 7.10.0 (R2010a). On each problem, the following three algorithms were analyzed and simulated.

- 1) The conventional Back Propagation Gradient Descent (BPGD)
- 2) The Back Propagation Gradient Descent with Adaptive Gain (BPGD-AG) [14]
- 3) The proposed algorithm which is Back Propagation Gradient Descent with Adaptive Gain, Adaptive Momentum and Adaptive Learning Rate (BPGD-AGAMAL)

To compare the performance of the proposed algorithm with conventional BPGD and BPGD-AG [14], network parameters such as network size and architecture (number of nodes, hidden layers and so forth), values for the initial weights and gain parameters were kept the same. For all problems, the NN had one hidden layer with five hidden nodes and sigmoid activation function was used for all nodes. All algorithms were tested using the same initial weights which were randomly initialised from range [0,1] and received the input patterns for training in the same sequence.

For all training algorithms, as the gain, MC and LR value were modified; the weights and biases were updated using the new value of gain, MC and LR. To avoid oscillations during training and to achieve convergence, an upper limit of 1.0 is set for the gain value. The initial value used for the gain parameter is set to one. The initial value for MC and LR is randomly generated depends on the dataset problems. For each run, the numerical data is stored in two files - the results file and the summary file. The result file lists the data about each network. The number of iterations until the network converged is accumulated for each algorithm which is the mean, the standard deviation (SD) and the number of failures is calculated. The networks that failed to converge are obviously excluded from the calculations of the mean and SD and were considered to be reported as failures. For each problem, 50 different trials were run, each with different initial random set of weights. For each run, the number of iterations required for convergence is reported. For an experiment of 50 runs, the mean of the number of iterations (mean), the SD, and the number of failures are collected. A failure occurs when the network exceeds the maximum iteration limit; each experiment is run to 10 000 iterations; otherwise, it is halted and reported as a failure. Convergence is achieved when the outputs of the network conform to the error criterion as compared to the desired outputs.

4.1 Glass Classification Problem

This dataset was collected by B. German on fragments of glass encountered in forensic work. The glass dataset is used for separating glass splinters into six classes, namely float processed building windows, non-float processed building windows, vehicle windows, containers, tableware, or head lamps [15]. The selected architecture of the network is 9-5-6 with target error was set to 0.001. The best MC and LR value for conventional BPGD and BPGD-AG for the glass dataset are 0.1 and 0.1 while BPGD-AGAMAL is initialized randomly in range $[0.1,0.3]$ for MC and $[0.1,0.2]$ for LR value.

Table 1. Algorithm performance for Glass Classification Problem [15]

	BPGD	BPGD-AG	BPGD-AGAMAL
Mean	8613	2057	2052
Total CPU time (s) of converge	572.54	59.57	56.16
CPU time(s)/Epoch	6.65×10^{-2}	2.9×10^{-2}	2.74×10^{-2}
SD	2.15×10^3	2.45×10	3.12×10
Accuracy (%)	79.42	79.98	82.24
Failures	35	0	0

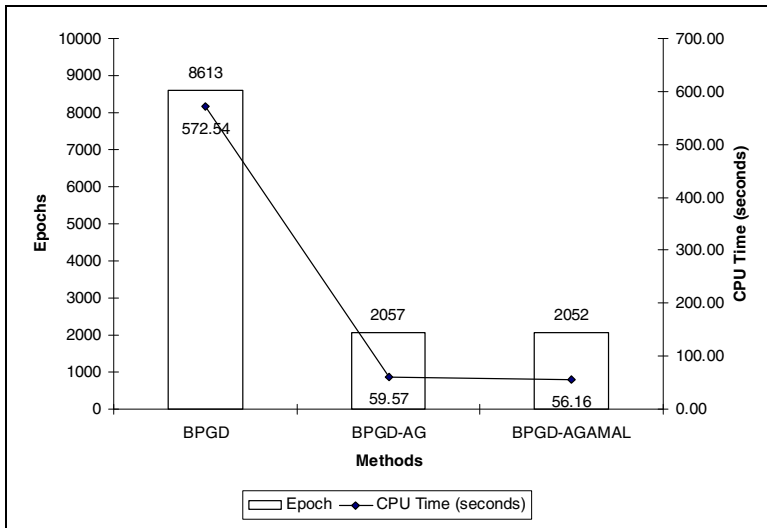


Fig. 1. Performance comparison of BPGD-AGAMAL with BPGD-AG and conventional BPGD on Glass Classification Problem

Table 1 shows that the proposed algorithm (BPGD-AGAMAL) exhibit excellent average performance in order to reach the target error. Furthermore, the accuracy of the proposed algorithm is better compared to BPGD and BPGD-AG. Moreover, the proposed algorithm (BPGD-AGAMAL) needs 2052 epochs to converge as opposed to

the conventional BPGD at about 8613 epochs, while BPGD-AG needs 2057 epochs to converge. Apart from speed of convergence, the time required for training the classification problem is another important factor when analyzing the performance. For numerous models, training process may suppose as a very important time consuming process. The graph depicted in Fig. 1 clearly show that the proposed algorithm (BPGD-AGAMAL) practically outperformed conventional BPGD with an improvement ratio, 10.2 seconds whilst BPGD-AG, the proposed algorithm outperformed with an improvement ratio nearly 2 seconds for the total time of converged. Besides, the BPGD did not perform well in this dataset since 70% of simulation results failed in learning the patterns.

4.2 Soybean Classification Problem

The soybean data set was constructed to classify 19 different diseases of soybeans. The discrimination is done based on a description of the bean (e.g. whether its size and color are normal or not) and the plant (e.g. the size of spots on the leaves, whether these spots have a halo, whether plant growth is normal whether roots are rotted or not) and also information regarding the history of the plant's life (e.g. whether changes in crop occurred in the last year or last two years, whether seeds were treated or not, the effect of the temperature environment). The selected architecture of the network is 35-5-19 and the target error was set as 0.001. The best MC for conventional BPGD and BPGD-AG is 0.1, meanwhile the best LR value for the soybean dataset is 0.1 and 0.4. The MC value for BPGD-AGAMAL is initialised randomly in range $[0.1,0.2]$ for MC and $[0.3,0.6]$ for LR value.

Table 2. Algorithm performance for Soybean Classification Problem [16]

	BPGD	BPGD-AG	BPGD-AGAMAL
Mean	3038	1271	1089
Total CPU time (s) of converge	311.47	91.92	78.63
CPU time(s)/Epoch	1.02×10^{-1}	7.23×10^{-2}	7.22×10^{-2}
SD	3.38×10^3	1.92×10^2	8.58×10
Accuracy (%)	94.23	91.08	94.82
Failures	8	0	0

Fig. 2 proved that the proposed algorithm (BPGD-AGAMAL) still outperformed other algorithms in terms of CPU time, number of epochs and accuracy. The proposed algorithm required 1089 epochs in 80.83 seconds CPU times to achieve the target error by 94.82% accurate. Whereas BPGD-AG required 1271 epochs in 91.92 seconds CPU times with 91.08% accurate. At the same time, BPGD needs 3038 epochs in 311.47 seconds CPU times and 94.23% accurate. As we can see in Table 2, the average number of learning iterations for the BPGD-AGAMAL was reduced up to 2.8 and 1.2 faster as compared to BPGD and BPGD-AG.

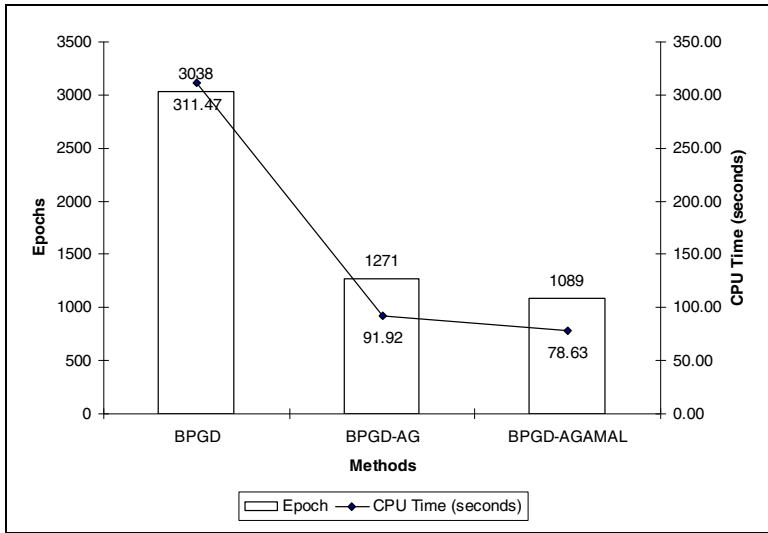


Fig. 2. Performance comparison of BPGD-AGAMAL with BPGD-AG and conventional BPGD on Soybean Classification Problem

4.3 Breast Cancer Classification Problem

This dataset was generated from University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg [17]. The input attributes are for instance the clump thickness, the uniformity of cell size, the uniformity of cell shape, the amount of marginal adhesion, the single epithelial cell size, frequency of bare nuclei, bland chromatin, normal nucleoli and mitoses. This problem tries to diagnosis of Wisconsin breast cancer by trying to classify a tumor as either benign or malignant based on cell description gathered by microscopic examination. The selected architecture of the network is 9-5-2 with target error 0.001. The best MC for conventional BPGD and BPGD-AG for the breast cancer dataset is 0.1 and LR is 0.4 whilst BPGD-AGAMAL is randomly initialized in range of $[0.3, 0.6]$ for MC and $[0.1, 0.2]$ for LR value.

Table 3. Algorithm performance for Breast Cancer Classification Problem [17]

	BPGD	BPGD-AG	BPGD-AGAMAL
Mean	3136	590	526
Total CPU time (s) of converge	128.13	14.43	12.44
CPU time(s)/Epoch	4.09×10^{-2}	2.45×10^{-2}	2.37×10^{-2}
SD	1.95×10^3	2.63×10^2	3.12×10
Accuracy (%)	68.29	94.12	95.47
Failures	0	0	0

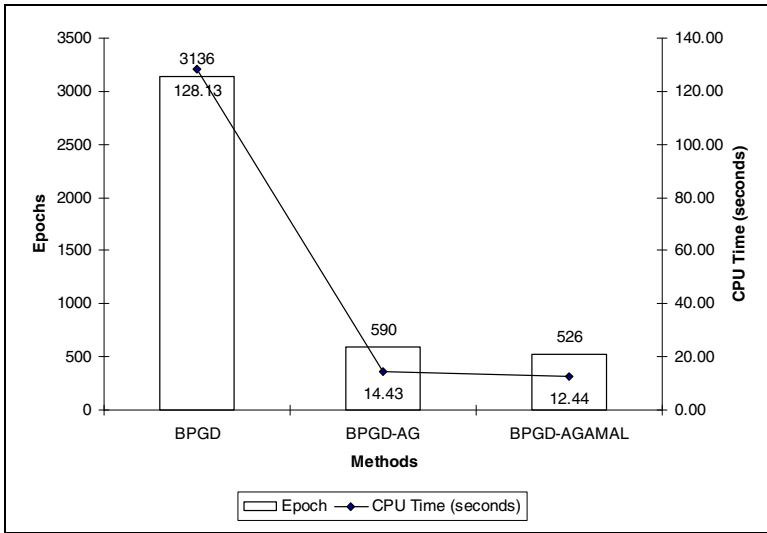


Fig. 3. Performance comparison of BPGD-AGAMAL with BPGD-AG and conventional BPGD on Breast Cancer Classification Problem

From Fig.3, it is worth noticing that the performance of the BPGD-AGAMAL is 83.23% faster than BPGD and almost 10.9% faster than BPGD-AG. Table 3 reveals that BPGD-AGAMAL approximately took 2.37×10^{-2} per epoch to reach target error as well as 95.47% accurate. While, BPGD-AG took 2.45×10^{-2} per epoch to reach target error with 94.12% accurate and BPGD took 4.09×10^{-2} per epoch to reach target error by 68.29% accurate. Still, the proposed algorithm (BPGD-AG) surpasses the BPGD and BPGD-AG algorithm in terms of total time of converge and accuracy to learn the pattern.

The results show that the BPGD-AGAMAL perform considerably better as compared to BPGD and BPGD-AG. Moreover, when comparing the proposed algorithm with BPGD and BPGD-AG, it has been empirically demonstrated that the proposed algorithm (BPGD-AGAMAL) performed highest accuracy than BPGD and BPGD-AG algorithm. This conclusion enforces the usage of the proposed algorithm as an alternative training algorithm of BP algorithm.

5 Conclusions

Although back propagation algorithm is widely implemented in the most practical NN applications and performed relatively well, this algorithm still needs some improvements. We have proposed a further improvement on the current working algorithm proposed by Nazri *et al.* [14]. The proposed algorithm adaptively changes the gain parameter of the activation function together with MC and LR to hasten up the convergence behaviour as well as slide the network through shallow local minima. The effectiveness of the proposed algorithm has been compared with the conventional Back Propagation Gradient Descent (BPGD) and Back Propagation Gradient Descent

with Adaptive Gain (BPGD-AG) [14]. The three algorithms had been verified by means of simulation on three classification problems including glass dataset with an improvement ratio 10.2 seconds for the BPGD and nearly 2 seconds better for the BPGD-AG in terms of total time to converge. Meanwhile, for soybean dataset, BPGD-AGAMAL was reduced up to 2.8 and 1.2 faster as compared to BPGD and BPGD-AG. While breast cancer dataset indicates that BPGD-AGAMAL is 83.23% faster than BPGD and almost 10.9% faster than BPGD-AG respectively. The results show that the proposed algorithm (BPGD-AGAMAL) has a better convergence rate and learning efficiency as compared to conventional BPGD and BPGD-AG [14].

Acknowledgments. The authors would like to thank **Universiti Tun Hussein Onn Malaysia (UTHM)** for supporting this research under the Postgraduate Incentive Research Grant.

References

1. Haykin, S.: *Neural Networks and Learning Machines*. Prentice Hall, New Jersey (2009)
2. Nawi, N.M., Ransing, R.S., Salleh, M.N.M., Ghazali, R., Hamid, N.A.: An Improved Back Propagation Neural Network Algorithm on Classification Problems. In: Zhang, Y., Cuzzocrea, A., Ma, J., Chung, K.-i., Arslan, T., Song, X. (eds.) DTA and BSBT 2010. *Communications in Computer and Information Science*, vol. 118, pp. 177–188. Springer, Heidelberg (2010)
3. Nawi, N.M., Ghazali, R., Salleh, M.N.M.: The Development of Improved Back-Propagation Neural Networks Algorithm for Predicting Patients with Heart Disease. In: Zhu, R., Zhang, Y., Liu, B., Liu, C. (eds.) ICICA 2010. *LNCS*, vol. 6377, pp. 317–324. Springer, Heidelberg (2010)
4. Sabeti, V., Samavi, S., Mahdavi, M., Shirani, S.: Steganalysis and Payload Estimation of Embedding in Pixel Differences using Neural Networks. *Pattern Recogn.* 43, 405–415 (2010)
5. Mandal, S., Sivaprasad, P.V., Venugopal, S., Murthy, K.P.N.: Artificial Neural Network Modeling to Evaluate and Predict the Deformation Behavior of Stainless Steel Type AISI 304L during Hot Torsion. *Applied Soft Computing* 9, 237–244 (2009)
6. Subudhi, B., Morris, A.S.: Soft Computing Methods Applied to the Control of a Flexible Robot Manipulator. *Applied Soft Computing* 9, 149–158 (2009)
7. Lee, K., Booth, D., Alam, P.: A Comparison of Supervised and Unsupervised Neural Networks in Predicting Bankruptcy of Korean Firms. *Expert Systems with Applications* 29, 1–16 (2005)
8. Sharda, R., Delen, D.: Predicting Box-Office Success of Motion Pictures with Neural Networks. *Expert Systems with Applications* 30, 243–254 (2006)
9. Yu, L., Wang, S.-Y., Lai, K.K.: An Adaptive BP Algorithm with Optimal Learning Rates and Directional Error Correction for Foreign Exchange Market Trend Prediction. In: Wang, J., Yi, Z., Žurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006. *LNCS*, vol. 3973, pp. 498–503. Springer, Heidelberg (2006)
10. Lera, G., Pinzolas, M.: Neighborhood Based Levenberg-Marquardt Algorithm for Neural Network Training. *IEEE Transaction on Neural Networks* 13, 1200–1203 (2002)
11. Wang, X.G., Tang, Z., Tamura, H., Ishii, M., Sun, W.D.: An Improved Backpropagation Algorithm to Avoid The Local Minima Problem. *Neurocomputing* 56, 455–460 (2004)

12. Otair, M.A., Salameh, W.A.: Speeding Up Back-Propagation Neural Networks. In: Proceedings of the 2005 Informing Science and IT Education Joint Conference. pp.167-173. Flagstaff, Arizona, USA (2005)
13. Ji, L., Wang, X., Yang, X., Liu, S., Wang, L.: Back-Propagation Network Improved by Conjugate Gradient Based on Genetic Algorithm in Qsar Study on Endocrine Disrupting Chemicals. Chinese Science Bulletin 53, 33–39 (2008)
14. Nawi, N.M., Ransing, R.S., Ransing, M.S.: An Improved Conjugate Gradient Based Learning Algorithm for Back Propagation Neural Networks. International Journal of Information and Mathematical Sciences 4, 46–55 (2008)
15. Evett, I.W., Spiehler, E.J.: Rule Induction in Forensic Science. Knowledge Based Systems, 152–160 (1988)
16. Michalski, R.S., Chilausky, R.L.: Learning by Being Told and Learning from Examples: An Experimental Comparison of the Two Methods of Knowledge Acquisition in the Context of Developing an Expert System for Soybean Disease Diagnosis. International Journal of Policy Analysis and Information Systems 4(2) (1980)
17. Mangasarian, O.L., Wolberg, W.H.: Cancer diagnosis via linear programming. SIAM News 23, 1–18 (1990)
18. Ye, Y.C.: Application and Practice of the Neural Networks. Scholars Publication, Taiwan (2001)
19. Maier, H.R., Dandy, G.C.: The Effect of Internal Parameters and Geometry on The Performance Of Back-Propagation Neural Networks: An Empirical Study. Environmental Modelling and Software 13, 193–209 (1998)
20. Thimm, G., Moerland, P., Fiesler, E.: The Interchangeability of Learning Rate and Gain in Backpropagation Neural Networks. Neural Comput. 8, 451–460 (1996)
21. Eom, K., Jung, K., Sirisena, H.: Performance Improvement of Backpropagation Algorithm by Automatic Activation Function Gain Tuning Using Fuzzy Logic. Neurocomputing 50, 439–460 (2003)
22. Hamid, N.A., Nawi, N.M., Ghazali, R.: The Effect of Adaptive Gain and Adaptive Momentum in Improving Training Time of Gradient Descent Back Propagation Algorithm on Classification Problems. In: Proceeding of the International Conference on Advanced Science, Engineering and Information Technology, Hotel Equatorial Bangi-Putrajaya, Malaysia , pp. 178–184 (2011)

Author Index

- A. Al-Jamimi, Hamdi III-611
A. Aljunid, S. III-728
A. Jabar, Marzanah I-46, III-586
A. Rashid, M. III-728
A. Tønnesland, Trygve III-163
Ab Manan, Jamalul-lail III-408, II-376,
II-437
Abbas, Mohammed I-517, II-310
Abbas, Wan Faezah I-527
Abdalla, Ahmed I-600
Abd Ghani, Abdul Azim II-125
Abd Khalid, NoorElaiza II-556
Abd. Jalil, Kamarularifin III-408
Abd Rahman, Noor Ardian I-527
Abd Wahab, Mohd Helmy II-244,
III-109
Abd. Wahid, Mohamad Halim II-567
Abdelhaq, Maha III-429
Abd Jalil, Kamarularifin II-437
Abdul Hamid, Norhamreeza III-812
Abdul Khalid, Noor Elaiza I-660
Abdul Majid, Mohd. Shahmi II-672
Abdul Manaf, Azizah I-674
Abdul Rejab, Mohamad Rizal III-398
Abdullah, Abdul Hanan I-401
Abdullah, Azizol I-416, III-419
Abdullah, Azrai III-571
Abdullah, Azween I-448
Abdullah, Junaidi I-623
Abdullah, Rusli I-78, II-125, II-346
Abdullah, Zailani II-480, II-495
Abdullah, Zul Hilmi II-177
Abdul Manaf, Azizah II-310
Abro, Abdul Ghani I-353
Abu Bakar, Kamalrulnizam I-401
Adhipta, Dani II-652
Adnan Fadil, Hilal III-728
Affandy III-557
Afsar, Sadia III-755
Ahmad, Adnan II-335
Ahmad, Idawaty II-618
Ahmad, Kamsuriah III-600
Ahmad, Mohd Yamin I-652
Ahmad, Nurul Haszeli II-376
Ahmad, Rohana II-556
Ahmad, Sabrina II-724, III-542
Ahmad, Siti Arpah II-556
Ahmad, Zaid III-408
Ahmad Dahlan, Abdul Rahman II-254
Ahmad Fauzi, Mohammad Faizal I-723
Ahmed, Moataz III-25, III-735
Ahmed, Nasim III-728
Ahmed Ali, Ghassan III-777
Akama, Kiyoshi III-702
Akbar, Habibullah I-747
Akbar, Rehan III-571
Albouraey, Fathy I-1
Aleksieva-Petrova, Adelina I-463
Al-Herz, Ahmed III-25
Ali Othman, Zulaiha III-274
Alizadeh, Hadi III-533
Aljunid, Syed Ahmad II-376
Al-Mansoori, Mohammed Hayder
II-672
Almomani, Omar III-717
Al-Obaidi, Alaa M. III-463
Alsaggaf, Ebtehal I-1
Alsaqour, Raed III-429
Alshayeb, Mohammad III-611
Alshehri, Saleh I-332
Al-Shqeerat, Khalil II-96
Al-Shrouf, Faiz II-96
A. Manaf, Azizah I-592
Amin, Aamir III-205
Amir Sjarif, Nilam Nur I-687
Anam, Rajibul I-244
Anas Bassam, AL-Badareen I-46,
III-586
Anwar, Fares III-600
Arbaiy, Nureize II-81
Arfan Wardhana, Andrianto III-306
Arif, Fahim III-755
Asgari, Mojtaba III-384
Ashraf, Mahmood III-231
Asnawi, Rustam I-111
Asyrani Sulaiman, Hamzah III-493
Atan, Rodziah II-125
Aungsakun, Siriwadee II-714

- Awad, Ali Ismail I-122
 Awang, Zaiki I-332
 Awang Rambli, Dayang Rohaya I-292
 Azzali, Fazli III-398
- B. Ahmad, Rohiza III-291, III-728
 B.C. Haron, Mohammad III-455
 Baba, Kensuke I-122
 Baba, Mohd Sapiyan III-257
 Bade, Abdullah III-493
 Bagheri, Alireza I-441
 Baharom, Fauziah I-133
 Baharudin, Baharum I-317
 Bahiyah Rahayu, Syarifah III-306
 Bajec, Marko I-232
 Basaruddin, Suzana II-43
 Basri, Shuib II-214
 Bekiarov, Luka I-463
 Bin Mahmood, Ahmad Kamil II-167
 Bin Mat Nor, Fazli II-437
 Bin Sulong, Ghazali I-567
 bin Masri, Syafrudin I-353
 Binti Latip, Rohaya I-709
 Binti Salam, Sazilah III-557
 Binti Zulkefli, Sufia Ruhayani I-111
 Borovska, Plamenka I-463
 Bozorgi, Mohammadmehdi III-793
 Bt Abdul Manaf, Azizah I-517
- C. Haron, M. Bakri I-168
 C.A. Gjerde, Njaal III-163
 Castanon-Puga, Manuel I-391, III-624
 Castro, Juan R. I-391
 Chang, Chun-Hyon III-119, III-266
 Che Fauzi, Ainul Azila II-244
 Che Pa, Noraini III-803
 Chhillar, Rajender Singh III-659
 Chong, Mien May I-709
 Choo, Yun-Huoy I-637, II-590
 Chueh, Hao-En I-68
 Chuprat, Suriayati I-674
- Dahiya, Deepak III-502
 Dhillon, Sharanjit Kaur I-92
 Din, Jamilah I-46, III-586
 Dinca, Ionut III-76
 Draman, Noor Azilah II-724
- Elammari, Mohamed III-54
 El-Attar, Mohamed III-735
 El-Halees, Alaa II-107
- El-Qawasmeh, Eyas II-289
 El Toukhy, Ahmed III-443
 El-Zaart, Ali I-735
 Emami, Mir Shahriar I-567
 Embong, Abdullah II-28, II-694
- F. Alteap, Tarek III-54
 F.J. Klaib, Mohammad II-244, III-91,
 III-99
 Fahmideh, Mahdi III-631
 Fakharaldien, M.A. Ibrahim II-460
 FanJiang, Yong-Yi I-217
 Fazal-e-Amin III-66
 Farhang, Yousef II-680
 Farisi, Al I-269
 Fattahi, Haniyeh II-680
- Gani, Abdullah III-257
 Garcia-Valdez, Mario III-624
 Gaxiola-Vega, Luis Alfonso III-624
 Ghazali, Masitah III-231
 Ghazali, Rozaida II-530, III-812
 Gomai, Abdu I-735
 Gopalakrishnan, Sundar III-163
 Gutub, Adnan Abdul-Aziz I-104
- H. Beg, A. II-232
 H.M. Radzi, Muhammad III-455
 Hafizah, Siti III-257
 Haji-Ahmed, Ahmed Abdulaziz II-254
 HajYasien, Ahmed II-542
 Hamdan, Abdul Razak III-274
 Handaga, Bana II-575
 Haron, Haryani II-43
 Harrag, Fouzi II-289
 Hartwig, Michael III-15
 Hasbullah, Halabi III-368
 Hasan, Mohd Hilmi II-412
 Hassan, Mahamat Issa I-448
 Hassan, Md. Mahedi III-342
 Hassan, Mohd Fadzil I-269, II-361,
 II-412, II-652, II-663, III-176, III-571
 Hassan, Nor Hafeizah I-16
 Hassan, Rosilah III-429
 Hassan, Suhaidi III-717
 Haw, Su-Cheng I-723, III-130
 Hayat Khan, Razib I-31
 Heegaard, Poul E. I-31
 Hegadi, Rajendra II-427

- Herawan, Tutut II-1, II-16, II-137,
II-480, II-495, II-516
- Herman, Nanna Suryana III-557
- Hj Arshad, Noor Habibah I-210
- Hj. Yahaya, Jamaiah I-133
- Ho, Chin Kuan I-244
- Hoong, Poo Kuan III-342
- Huh, Eui-Nam III-324
- Husaini, Noor Aida II-530
- Hussin, Burairah I-16
- I. Sultan, E. II-232
- I. Younis, Mohammed III-1
- Ibrahim, Mohd Izham II-391
- Ibrahim, Rabi'u I-278
- Ibrahim, Rosziati II-516
- Ibrahim, Zuwairie I-144
- Idris, Zulkhairi II-688
- Ipate, Florentin III-76
- Iqbal, Muzaffar III-518
- Isa, Mohd Adham III-246, III-764
- Ismail, Habibah II-274
- Ismail, Idris II-448
- Ismail, Lokman Hakim II-530
- Ismail, Mahamod III-384
- Ismail, Zuraini I-517
- Jaafar, Azmi II-346
- Jaafar, Jafreezal III-176
- Jabeen, Shunaila III-143
- Jamain, N.S. I-506
- Jamshidi, Pooyan III-631
- Jangjou, Mehrdad I-441
- Janosepah, Safoura III-533
- Jantan, Adznan I-332
- Jantan, Aman I-199, II-58, II-391,
II-403, III-693, III-777
- Jaya, Hang Tuah I-547
- Jumari, Kasmiran I-698, III-384
- Jusoh, Nor Amizam I-581
- K. Subramaniam, Shamala II-618,
III-419
- Kadir, Rabiah Abdul II-225
- Kamalia Azma, Kamaruddin I-527
- Karimi Sani, Navid II-263
- Kasirun, Z.M. I-506
- Kateretse, Cecile III-324
- Kawamura, Takao II-298
- Ketabchi, Shokoofeh II-263
- Khalid, Marzuki I-144
- khan, Aurangzeb I-317
- khan, Khairullah I-317
- Khari, Manju III-336
- Khatib, Jawdat II-254
- Khatun, Sabira I-332, III-99
- Khorsandi, Siavash III-353
- Kim, Doo-Hyun III-119, III-266
- Kim, Tae-Wan III-119, III-266
- Kim, Yun-Kwan III-119
- Kiong Chi, Yip I-538
- Kochar, Barjesh III-659
- Koike, Hidekatsu III-702
- Kolivand, Hoshang II-680
- Ku-Mahamud, Ku Ruhana I-365
- Kuo, Jong-Yih I-217
- Latif, Norizal Abd II-412
- Latip, Rohaya I-416, II-688
- Lau, Bee Theng I-486
- Lavbič, Dejan I-232
- Lee, Chien-Sing III-130
- Lee, Ho Cheong II-627
- Lee, Sai Peck III-463
- Lefticaru, Raluca III-76
- Lerthathairat, Pornchai III-478
- Liew, Siau-Chuin I-555
- Lim, Chin Siong I-342
- Lim, Tek Yong I-244
- Limsakul, Chusak II-703, II-714
- Lin, Yo-Hsien I-68
- Liu, Kecheng II-263
- Lotfi, Shahriar II-605
- Low, Boon Wee III-151
- Lu, Ta-Te I-612
- M. Abdullallah, Radhwan III-419
- M. Abdelaziz, Tawfig III-644
- M. Azmi, Zafril Rizal I-401
- M. Eide, Magnus III-163
- M. Lazim, Yuzarimi II-116
- M. Monzer Habbal, Adib III-398
- M. Zeki, Akram I-592
- Ma, Shang-Pin I-217
- Ma, Xiuqin I-259, II-1, II-16, II-148,
II-642
- Maarof, Mohd Aizaini III-793
- Machfud, Alfi Khairiansyah II-254
- Madi, Mohammed III-717

- Magdaleno-Palencia, Jose Sergio III-624
- Magdaleno-Palencia, Sergio I-391
- Mahamat Pierre, Djamalladine II-205
- Mahdi, Khaled III-443
- Mahjoub, Shabnam II-605
- Mahmod, Ramlan II-177
- Mahmood Bin, Ahmad Kamil I-425, II-652, III-66, III-205, III-677
- Mahmud, Rozi I-332
- Mahmud, Zamalia I-168
- Makina, T. I-179
- Mamat, Rabiei II-137
- Manaf, Mazani I-547, II-335
- Mansor, Zulkefli I-210
- Márquez, Bogart Yail I-391, II-509
- Masood, Iram III-143
- Masroor Ahmed, Mohammad I-600, II-99
- Matayong, Sureena II-167
- Mat Deris, Mustafa II-137, II-480, II-575
- Mat Deris, Sufian II-116
- Mat Using, Siti Noradlina III-291
- Mateen, Abdul III-755
- Matsuzaki, Takanori II-298
- Mehmood Shah, Syed Nasir I-425, III-677
- Md Ali, Asma III-41
- Md. Ali, Siti Aishah I-652
- Md. Din, Norashidah II-672
- Md Ariff, Norharyati I-660
- Md Norwawi, Norita I-365
- Md Said, Abas II-448
- MD Sharif, Syahrizal Azmir I-401
- Mine, Tsunenori II-298
- Mir, Nighat I-306
- Mohd. Sidek, Roslina II-244, III-109
- Modiri, Nasser II-321, III-533
- Mohamad, Radziah III-283
- Mohamed, Azlinah I-652
- Mohamed, Farham II-116
- Mohamad Noor, Noor Maizura III-274
- Mohamed Noor, Noorhayati I-660
- Mohamad Noor, Norzaliha I-78
- Mohamad Saleh, Junita I-353
- Mohamad Zain, Jasni I-342, I-555, I-567, I-581, I-600, II-1, II-16, II-28, II-460, II-470, II-694
- Mohamed Salleh, Faridah Hani III-191
- Mohamed Zabil, Mohd Hazli III-1
- Mohamud Sharif, Abdullahi II-214
- Mohamad Yusop, Nor Shahida I-527
- Mohammadi Noudeh, Abdolrahman III-533
- Mohan Garg, Rachit III-502
- Mohd Hashim, Siti Zaiton I-687
- Mohd Nawi, Nazri I-380, II-516, II-530, III-812
- Mohd Salleh, Mohd Najib III-812
- Mohd Shapri, Ahmad Husni II-567
- Mohd. Taib, Shakirah III-291
- Mohd Yusof, Yasmin Anum I-652
- Mohd Yusoff, Siti Kamaliah II-448
- Mohd Zaini, Khuzairi III-398
- Mohd Zaki, Mohd Zulkifli III-246
- Mohd Zin, Abdullah III-803
- Mohd Zin, Noriyani II-232
- Mohamad, Rosmayati III-274
- Muda, Azah Kamilah I-637, II-590, II-724
- Muda, Noor Azilah I-637, III-542
- Mughal, Bilal Munir III-368
- Mushtaq, Arif I-292
- Mustapha, Muhazam III-215
- Muthuraman, Sangeetha III-91
- N. Abd Alla, Ahmed II-232
- N.A. Jawawi, Dayang N.A. II-274, III-246, III-764
- Nakov, Ognian I-463
- Nantajeewarawat, Ekawit III-702
- Ng, Choon-Ching II-157
- Ng, Hu I-623, I-723
- Nik Daud, Nik Ghazali III-215
- Noah, Shahrul Azman III-306
- Noersasongko, Edi III-557
- Noraziah, Ahmed II-232, II-244, III-91, III-109
- O. Elish, Mahmoud III-611
- Ocegueda-Hernández, Juan Manuel II-509
- Ørbekk, Kjetil III-163
- Omar, Nasiroh III-455
- Omar, Nazlia II-225
- Ong, Soon-Chuan I-144
- Oskooyee, Koosha Sadeghi I-441
- Othman, Mohamed I-56, I-416, II-618
- Oxley, Alan I-278, I-425, II-73, III-66, III-677

- Palaniappan, Sellappan I-538
 Pavlovich, Vladimir I-144
 Pedrycz, Witold II-190
 Periasamy, Elango I-188
 Phinyomark, Angkoon II-703, II-714
 Phothisonothai, Montri II-703
 Phukpattaranont, Pornchai II-703,
 II-714
 Pratama, Satrya Fajri I-637
 Pratiwi, Lustiana II-590
 Profitis, Anastasios I-463
 Prompoon, Nakornthip III-478
 Purnami, Santi Wulan II-694

 Qin, Hongwu II-1, II-16
 Qureshi, Barketullah I-56
 Qureshi, Muhammad Aasim II-663

 R. Othman, Rozmie III-1
 Rabbi, Khandakar Fazley III-99
 Radman, Abduljalil I-698
 Radzi, Nurul Asyikin Mohamed II-672
 Rahman, M. Nordin A. II-116
 Rahman, Norazeani Abdul II-567
 Raja Abdullah, R.S.A. I-332
 Ramli, Azizul Azhar II-190
 Rani, Mamta I-259, II-642
 Ranjbar, Leila III-353
 Rasmi, M. II-403
 Razali, Rozilawati III-600
 Rehman, Mobashar III-205
 Rehman, M.Z. I-380
 Riahi Kashani, Mohammad Mansour
 I-441
 Riyadi Yanto, Iwan Tri II-516
 Riaz, Sadia I-292
 Richardson, Joan III-41
 Rizwan, Muhammad III-518
 Romli, Rahiwan Nazar I-401
 Romli, Rohaida I-471

 S. Al-Salman, Abdul Malik I-735
 S. Mahmud, Shayma'a I-592
 Saari, Eviyanti III-693
 Sabeil, Elfadil I-517, II-310
 Sabraminiam, Shamala I-56
 Sadeh, Ben III-163
 Safar, Maytham III-443
 Safdar, Sohail II-361
 Safei, Suhailan II-116

 Sahib, Shahrin I-16, I-747
 Saini, Radhika III-336
 Salam, Sazilah I-179
 Salman Al-Salman, Abdul Malik II-289
 Saleem, Muhammad Qaiser III-176
 Saliimi Lokman, Abbas II-470
 Salleh, Rohani I-292, III-205
 Sammi, Rabia III-143
 Samsudin, Khairulmizam II-177
 Santoso, Heru-Agus III-130
 Selamat, Ali II-157
 Selamat, Mohd Hasan I-46, I-78, II-346,
 III-586
 Selamat, Siti Rahayu I-16
 Sembiring, Rahmat Widia II-28
 Sembok, Tengku Mohd. Tengku II-225
 Senan, Norhalina II-516
 Shams, Fereidoon III-631
 Shamsir, Mohd Shahir I-401
 Shamsuddin, Siti Mariyam I-687
 Shapiai, Mohd Ibrahim I-144
 Shibghatullah, Abdul Samad I-547
 Shin, Won III-266
 Sia Abdullah, Nur Atiqah II-346
 Sim, Yee-Wai I-155
 Stefanescu, Alin III-76
 Suarez, E. Dante I-391
 Sugahara, Kazunori II-298
 Suklaead, Pornpana II-703
 Sulaiman, Norrozila I-259, II-1, II-16,
 II-148, II-460, II-642
 Sulaiman, Shahida I-471
 Sunar, Mohd Shahrizal II-680
 Suryana, Nanna I-747
 Susilawati, Mohamad Fatma I-674
 Syu, Yang I-217

 Tafti, Ahmad Pahlavan III-533
 Taib, Haslina II-556
 Taib, Mohd Nasir II-556
 Takahashi, Kenichi II-298
 Tan, Ding-Wen I-155
 Tan, Wooi-Haw I-623
 Tao, Hai I-600
 Tarawneh, Feras I-133
 Tong, Hau-Lee I-623, I-723
 Tsai, Tsung-Hsuan I-612
 Tudose, Cristina III-76
 Turaev, Sherzod I-46, III-586
 Turani, Aiman II-96

- Udzir, Nur Izura II-177
- Valdés-Pasarón, Sergio II-509
- Valeria, Nia I-486
- Vali, Nasser II-321
- Varathan, Kasturi Dewi II-225
- Vitasari, Prima II-495
- Wagan, Asif Ali III-368
- Wajahat Kamal, Mohammed III-735
- Wan Ahmad, Wan Fatimah I-111
- Wan Hussin, Wan Muhammad Syahrir I-342
- Wan Ishak, Wan Hussain I-365
- Wan Mohd, Wan Maseri II-244
- Watada, Junzo II-81, II-190
- Wati, Nor Asila I-56
- Wati Abdul Hamid, Nor Asilah III-419
- Wen Jau, Lee I-144
- Wong, Chee Siang III-151
- Wu, Jia-Yuan I-612
- Yaakob, Razali II-125
- Yaakub, Che Yahaya III-99
- Yahaya, Bakri I-416
- Yahaya, Nor Adnan I-538
- Yahya, Saadiah I-210, I-660
- Yaik Ooi, Boon III-151
- Yeoh, William I-155
- Yongho Kim, Ronny III-313
- Yuhaniz, Siti Sophiayati I-687
- Yunus, Yuzaimi II-335
- Yusob, Bariah I-342
- Yusof, M. Kamir II-116
- Yusof, Yuhanis III-717
- Yusoff, Mohd Najwadi II-58
- Zainal Abidin, Siti Zaheera I-168, III-455
- Zainal Abidin, Zaheera I-547
- Zain, Zuhaira Muhammad II-125
- Zainal, Nasharuddin I-698
- Zakaria, Nordin II-205
- Zaman, Halimah Badioze I-188
- Zamin, Norshuhani II-73
- Zamzuri, Zainal Fikri II-335
- Zeshan, Furkh III-283
- Zhi Sam, Lee III-793
- Zolkipli, Mohamad Fadli I-199
- Zuhairi Zamli, Kamal I-471, III-1
- Zulkarnain, Zuriati II-618