Studies in Computational Intelligence 413

Roger Lee (Ed.)

Software and Network Engineering

Springer

Studies in Computational Intelligence 413

Editor-in-Chief

Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

For further volumes:
http://www.springer.com/series/7092

Roger Lee (Ed.)

Software and Network Engineering

Editor
Roger Lee
Central Michigan University
USA

ISSN 1860-949X
ISBN 978-3-642-28669-8
DOI 10.1007/978-3-642-28670-4
Springer Heidelberg New York Dordrecht London

e-ISSN 1860-9503
e-ISBN 978-3-642-28670-4

Library of Congress Control Number: 2012935659

Printed on acid-free paper

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

Preface

The purpose of the first ACIS International Symposium on Software and Network Engineering held on December 19–20, 2012 on the Seoul National University campus, Seoul, Korea is to bring together scientist, engineers, computer users, students to share their experiences and exchange new ideas, and research results about all aspects (theory, applications and tools) of software and network engineering, and to discuss the practical challenges encountered along the way and the solutions adopted to solve them. The symposium organizers selected the best 12 papers from those papers accepted for presentation at the symposium in order to publish them in this volume. The papers were chosen based on review scores submitted by members of the program committee, and underwent further rigorous rounds of review.

In Chapter 1, First, this paper evaluates the impression of questions and answers at Questions and Answers (Q & A) sites in order to avoid the problem of mismatch between the questioner and the respondent. Fifty impression words effective in evaluating impressive expression of statements are selected from a dictionary. An impressive evaluation experiment is then conducted for sixty questions and answers posted at Yahoo! Chiebukuro by using those impression words. Nine factors are obtained by applying factor analysis to the scores obtained through the experiment. Then factor scores of any other statements are tried to be estimated by using multiple regression analysis. This result, however, shows that the estimation accuracy is insufficient. To improve the estimation accuracy, the multiple regression analysis considering quadratic terms is applied. The result of the analysis shows that the estimation accuracy can be improved.

In Chapter 2, We propose an integrated hierarchical temporal memory (IHTM) network for continuous multi-interval prediction (CMIP) based on the hierarchical temporal memory (HTM) theory. The IHTM network is constructed by introducing three kinds of new modules to the original HTM network. One is Zeta1FirstNode which is used to cooperate with the original HTM node types for predicting stock price with multi-interval at any given time. The second is ShiftVectorFileSensor module used for inputting stock price data to the network continuously. The third is a MultipleOutputEffector module which produces multiple prediction results with different intervals simultaneously. With these three new modules, the IHTM

network make sure newly arriving data is processed and continuous multi-interval prediction is provided. Performance evaluation shows that the IHTM is efficient in the memory and time consumption compared with the original HTM network in CMIP.

In Chapter 3, As the Web becomes the major source for information and services, fast access to needed Web objects is a critical requirement for many applications.Various methods have been developed to achieve this goal. Web page prefetching is one of these methods that is commonly used and quite effective reducing the user perceived delays. In this paper, we proposed a new prefetching algorithm, called pART1, based on the original ART1 algorithm that is a neural network approach for clustering. We modified the ART1 algorithm to obtain 2-way weights (bottom-upand top-down) between the clusters of hosts and the URLs (Web pages), and use these weights to make prefetching decisions. In the experiments we conducted, the new algorithm outperformed the original ART1 in terms of cache hit ratio.

In Chapter 4, In this paper, we present a user-adaptive augmented reality (AR) system that augments physical objects with personalized content according to user's context as well as preferences. Since a user prefers different content according to the context, it reasons the user's recent content preferences through artificial neural networks trained with the feedback history describing which content the user liked or disliked with respect to his/her context. The system recommends a set of content relevant to the user's context and preferences. Then, it enables the user to select a preferred content among the recommended set and superimposes the selected content over physical objects. We implemented a prototype illustrating how our system could be used in daily life and evaluate its performance. From experimental results, we could confirm that our system effectively assisted users through personalized content augmentation in mobile computing environment.

In Chapter 5, Most of the software developed in universities is primarily used by the developers themselves. Normally, most of this software is managed and stored on servers in the research laboratories, but since the software is generally lacking in documentation and is not developed with third-party use in mind, it tends to be used only by the original developers. It is seldom reused after the developers have graduated, and is often not in a fit state for use by third parties. Today's information systems graduates have not been provided with sufficient education with regard to the knowledge, techniques and experience needed for the usual software development process of actual software development businesses from project planning through to execution and management (requirements analysis, development, implementation, testing, etc.) and lack the basic skills for handling actual business situations. In this paper, we report on our approach to software management using the UEC software repository to consolidate software assets, and on practical software development education based on this repository.

In Chapter 6, The Global Ad is a system to be used when users apply to publish their advertisement on newspapers. The advertisement is registered and uploaded on database that can be viewed by advertising agency. In actual use by multiple users, the system needs to allocate advertising spaces on the newspaper and to calculate

the cost to publish the advertisements. Also, the system need to solve a problem like a packing problem with publishing costs/bid value. To solve the problems, in this paper, we propose a winner determination method using 2-dimensional knapsack problem.

In Chapter 7, Network optimization is a classic and effective approach for allocating network resources in such a way that certain measure of the network utilization is optimized. Network models and algorithms have been proposed and developed for solving the optimization problems. However, we haven't seen studies on the effect of the utility functions on the network response time when the overall utilization of the network is maximized. In this paper, we investigate this problem with simulation experiments on a simple 4-node network using two different utility functions, a logarithmic function and a linear function. We fine tune the network transmission rates near their optimal values on several routes and observe the network response time. Our preliminary study showed that different utility functions do have impact on the response time on individual routes.

In Chapter 8, To detect features are significantly important for reconstructing a model in reverse engineering. In general, it is too difficult to find the features from the original industrial 3D CT data because the data have many noises. So it is necessary to reduce the noises for detecting features. This paper proposes a new method for detecting corner features and edge features from noisy 3D CT scanned data. First, we applied the level set method[18] to CT scanned image in order to segment the data. Next, in order to reduce noises, we exploited nonlocal means method[19] to the segmented surface. This helps to detect the edges and corners more accurately. Finally, corners and sharp edges are detected and extracted from the boundary of the shape. The corners are detected based on Sobel-like mask convolution processing with a marching cube. The sharp edges are detected based on Canny-like mask convolution with SUSAN method[13], which is for noises removal. In the paper, the result of detecting both features is presented.

In Chapter 9, This paper is intended to proposal lifecycle of open source software. There are many difficulty factors that cause the open source software problems during software interoperation. This paper evaluates the efficiency of lifecycle that detection of new risk items and remove ratio at the lifecycle of open source software.

In Chapter 10, This paper is intended to evaluate the risk items of lifecycle for the open source software. There are many difficulty factors that cause the open source software problems during software interoperation. Also, using defect cause, we understand associated relation between defects and design defect trigger. So when we archive correspond project, we can forecast defect and prepare to solve defect by using defect trigger. This paper evaluates the degree of risk of lifecycle that detection of new risk items and remove ratio at the lifecycle of open source software.

In Chapter 11, Advanced information technologies and the Internet have resulted in the emergence of the phenomenon of processing in the clouds (cloud computing - CC). A general definition of processing in the clouds is "access to a resource on the Internet outside the company firewall." This work is an introduction to a rapid development in web application functionality going beyond the traditional concept

of Web 2.0 applications. The new applications of cloud computing bring mobility and are delivered to all possible devices designed to interface with the user, ranging from PCs to smart phones. The aim of this review is to investigate the potential of these new solutions to combine the advantages of desktop applications (speed, ergonomics, user friendliness, access to local resources) and web applications (mobility, accessibility, scalability).

In Chapter 12, Mobile software development challenge the modelling activities that precede the technical design of a software system. The context of a mobile system includes a broad spectrum of technical, physical, social and organizational aspects. Some of these aspects need to be built into the software. Selecting the aspects that are needed is becoming increasingly more complex with mobile systems than we have previously seen with more traditional information systems. With great diversification in the software market a mobile embedded system is loaded with software from dozens of different vendors. With the wide variety of different mobile embedded systems applications, we need specific software for those specific mobile embedded systems that will meet the requirements of an application. In this paper, we the design and implements the procedures for mobile software architecture using Components Based Development(CBD) and object oriented methodology. It starts with the requirement analysis of a mobile embedded systems and continues to provide the functional model of the system and also the environmental design as well. As more and more functions or components needed to be added in designing a software system a modularized and systematic approach becomes not an option. This paper proposes a systematic and procedural approach that will help build a more reliable and suitable mobile software architecture. It is our sincere hope that this volume provides stimulation and inspiration, and that it will be used as a foundation for works yet to come.

Dec 2011 Roger Lee

Contents

List of Contributors

Yung-Cheol Byun
Jeju National University, Korea
E-mail: ycb@jejunu.ac.kr

Jianhua Diao
Dalian University of Foreign Languages, China
E-mail: software@dlufl.edu.cn

Wenying Feng
Trent University, Canada
E-mail: wfeng@trentu.ca

Sejin Oh
LG Electronics, Seoul, Korea
E-mail: sjin.oh@lge.com

Teruhisa Hochin
Kyoto Institute of Technology, Japan
E-mail: hochin@kit.ac.jp

Toufiq Hossain Kazi
Trent University, Canada
E-mail: kazihossain@trentu.ca

Gongzhu Hu
Central Michigan University, USA
E-mail: hu1g@cmich.edu

Hyun-Syug Kang
Gyeongsang National University, Korea
E-mail: hskang@gnu.ac.kr

Hiroki Nomiya
Kyoto Institute of Technology, Japan
E-mail: nomiya@kit.ac.jp

Tetsuji Satoh
University of Tsukuba, Japan
E-mail: satoh@slis.tsukuba.ac.jp

Yuya Yokoyama
Kyoto Institute of Technology, Japan
E-mail: is06067@yahoo.co.jp

Takaaki Goto
University of Electro-Communications, Japan
E-mail: goto@kikou.uec.ac.jp

Takahiro Homma
University of Electro-Communications, Japan
E-mail: homma@kikou.uec.ac.jp

Kensei Tsuchida
Toyo University, Japan
E-mail: kensei@toyo.jp

Tetsuro Nishino
University of Electro-Communications, Japan
E-mail: nishino@ice.uec.ac.jp

Satoshi Takahashi
University of Tsukuba, Japan
E-mail: takahashi2007@e-activity.org

Yuji Hashiura
Yamagata University, Japan
E-mail: hashiura2009@e-activity.org

Roger Y. Lee
Central Michigan University, USA
E-mail: lee1ry@cmich.edu

Tokuro Matsuo
Yamagata University, Japan
E-mail: matsuo@yz.yamagata-u.ac.jp

Chris Johns
Trent University, Canada
E-mail: christopherjohns@trent.ca

Kevin Mak
Trent University, Canada
E-mail: kevinmak@trent.ca

Thi-Chau Ma
Seoul National University, Korea
E-mail: ma.thi.chau@gmail.com

Chang-soo Park
Seoul National University, Korea
E-mail: winspark@snu.ac.kr

Kittichai Suthunyatanakit
Seoul National University, Korea
E-mail: skittichai@hotmail.com

Min-jae Oh
Seoul National University, Korea
E-mail: mjoh80@snu.ac.kr

Tae-wan Kim
Seoul National University, Korea
E-mail: taewan@snu.ac.kr

Myung-joo Kang
Seoul National University, Korea
E-mail: mkang@snu.ac.kr

Jan Seruga
Australian Catholic University
E-mail: jan.seruga@acu.edu.au

Ha Jin Hwang
Kazakhstan Institute of Management, Economics, and Strategic Research
E-mail: hjhwang@kimep.kz

Ji-Uoo Tak
Catholic University of Daegu, Korea
E-mail: lebbenle@cu.ac.kr

Haeng-Kon Kim
Catholic University of Daegu, Korea
E-mail: hangkon@cu.ac.kr

Eun-Ser Lee
Andong National University, South Korea
E-mail: eslee@andong.ac.kr

Joong-soo Kim
Andong National University, South Korea
E-mail: kimjs@andong.ac.kr

Obtaining Factors Describing Impression of Questions and Answers and Estimation of Their Scores from Feature Values of Statements

Yuya Yokoyama, Teruhisa Hochin, Hiroki Nomiya, and Tetsuji Satoh

Abstract. First, this paper evaluates the impression of questions and answers at Questions and Answers (Q & A) sites in order to avoid the problem of mismatch between the questioner and the respondent. Fifty impression words effective in evaluating impressive expression of statements are selected from a dictionary. An impressive evaluation experiment is then conducted for sixty questions and answers posted at Yahoo! Chiebukuro by using those impression words. Nine factors are obtained by applying factor analysis to the scores obtained through the experiment. Then factor scores of any other statements are tried to be estimated by using multiple regression analysis. This result, however, shows that the estimation accuracy is insufficient. To improve the estimation accuracy, the multiple regression analysis considering quadratic terms is applied. The result of the analysis shows that the estimation accuracy can be improved**.**

Keywords: Question & Answer site, Impression Evaluation, Factor Score, Multiple Regression Analysis, Quadratic term.

1 Introduction

In the Internet, an increasing number of people are using Questions and Answers (Q & A) sites recently [1, 2]. A Q&A site is a sort of communities where users

Yuya Yokoyama · Teruhisa Hochin · Hiroki Nomiya
Graduate School of Information Science, Kyoto Institute of Technology, Kyoto, Japan
e-mail: is06067@yahoo.co.jp, {hochin,nomiya}@kit.ac.jp

Tetsuji Satoh
Graduate School of Library, Information and Media Studies,
University of Tsukuba, Ibaraki, Japan
e-mail: satoh@slis.tsukuba.ac.jp

R. Lee (Ed.): Software and Network Engineering, SCI 413, pp. 1–13.
springerlink.com

post questions and answers mutually. At the same time, it is a site to solve various problems as well as to be used as the database with enormous knowledge. If a user posts a question, other users respond to it. A questioner selects a respond as the "Best Answer" that he or she has judged the most appropriate one, and he or she gives the respondent some points as a fee. The "Best Answer" is defined as a respond statement the questioner subjectively finds most satisfying. Several research efforts have been made to estimate the "Best Answer" [8-14].

With increasing users in the Q&A sites and more posted questions, it becomes harder for respondents to pick out appropriate questions suitable for their specialty and interest. Even if a user posts a question, it is neither necessarily seen nor given an answer by appropriate respondents. Moreover, mismatching caused by being unable to encounter an appropriate respondent may result in the following problems.

- A questioner may acquire wrong knowledge from inappropriate answers.
- Despite the answers, inappropriate knowledge makes impossible for respondents to answer the core of question, which eventually leaves the problem unsolved.
- People can be offended by answers including abusive words, slanders or statements against public order and standards of decency.

The end of this study is to introduce a question to the users who could appropriately answer it. This could avoid the problems described above. To this end, the correspondence between a question and its "Best Answer" is tried to be clarified. For clarifying the correspondence, the degrees of impression of statements are used. These degrees are experimentally obtained by evaluating the statements. First, impressive words used in the experiment are selected. The statements of questions and answers posted at Yahoo! Chiebukuro [2] are evaluated through the impressive words. By applying factor analysis to the scores obtained through the experiment, nine factors were obtained. By using the factor scores, the correspondence between a question and its "Best Answer" could be calculated.

The factor scores obtained, however, are only those statements used in the experiment, and others have not been obtained yet. In order to be able to estimate the factor scores of any other statements, multiple regression analysis is applied to the feature values of statements. We adopt word classes such as noun and verb, and the number of appearance or the percentage of Chinese and alphanumeric characters as feature values of statements. By considering quadratic terms, precision of estimation could be improved.

The remaining of this paper is as follows: Section 2 describes related works. Section 3 chooses impressive words used in the experiment. In Section 4, the impressions of the statements are experimentally evaluated. Section 5 shows the correspondence between a question and its "Best Answer." Section 6 describes feature values of sentences. Section 7 estimates the factor scores. Section 8 concludes the paper.

2 Related Works

Efforts of estimating the "Best Answers" have been made [3, 8-11]. Blooma *et al.* used non-textual features and textual ones to try to predict the "Best Answers" [8]. Agichtein *et al.* tried to assess the quality of questions and answers by using the content and usage features [9]. The analogical reasoning approach has been proposed [10]. This approach finds the "Best Answer" by using the links of questions and answers in the previous knowledge. Kim *et al.* have proposed the Best-Answer selection criteria [11]. For the information type questions, content values are important. Utility is important for the suggestion type questions, while socio-emotional values are important for the opinion type ones [11].

Nishihara *et al.* have proposed a method of detecting the answer statement that tends to be the "Best Answer" among the answers to a question [3]. By noticing the affinity of closing sentence expressions of questioners and respondents and clustering combinations of questions and the "Best Answers," they had a definite result. In their research, however, the proposed method focused on mere closing sentence expressions of questions and answers, not on contents of statements. Accordingly, we focused on impressive evaluation from style and contents of statements.

Kumamoto evaluated the impression from newspaper [4]. A hundred subjects read ten newspaper articles with forty-two impressive words by five levels (Agree, Fairly Agree, Fairly Disagree, Disagree, None), which were conducted ninth times. Collecting and analyzing impressive evaluation scores, he has proposed impressive axes suitable for expressing impression from newspaper. In this research, the target of the impressive evaluation experiment is a set of question and answer statements rather than newspaper articles.

3 Impressive Words

First, we extracted some words that seem effective as impressive words out of 21690 words listed in a vocabulary dictionary [5]. 806 words are extracted. Next, we chose impressive words according to the following procedures:

806 impressive words are classified into the following groups:

- Group A: Words used in positive sense
- Group B: Words used in negative sense
- Group C: Words that cannot be divided into A or B.

In each group, words of similar meanings are combined.

In evaluating sentences, the most generally used words are considered and determined.

Words of similar meanings are merged.

Through these procedures, 50 words are selected. Impressive words obtained are listed in Table 1. The words can be divided into words as to sentence styles and those as to sentence contents.

Table 1 Chosen Impressive Words

Expression of Statements(22 words)		Content of Statements(28 words)	
Easy	Persistent	Touching	Resentful
Skillful	Faltering	Wonderful	Disillusioning
Courteous	Dull	Favorable	Fearful
Beautiful	Insufficient	Impressive	Regrettable
Refreshing	Exaggerating	Accurate	Unjust
Fluent	Minute	Appropriate	Thoughtless
Special	Simple	Important	Amazing
Persuasive	Firm	Warm-Hearted	Real
Clear	Long	Creative	Inevitable
Ambiguous	Complicating	Fulfilling	Hot
Difficult	Original	Fun	Powerful
		Uncomfortable	Unexpected
		Suspicious	Marvelous
		Sharp	Dear

4 Experiments

4.1 Procedure

With using 50 impressive words, we conducted impressive evaluation experiment for 41 subjects (33 males and 8 females, age of 19-23). Experiment materials are twelve sets of question and answer statements (three each from four major categories: Yahoo! auction, PC & peripherals, love counseling & human relationships, and political & social problems), out of the statements actually posted at Yahoo! Chiebukuro [2] on Sep. 2005. Each set consists of a question and 4 answers (including the "Best Answer"), 60 statements are used in total.

The criteria of statement selection are as follows:

- Exclude too biased statements.
- Exclude statements contrary to public order and standards of decency.
- Exclude questions necessary to refer to URL.
- Select evenly the amounts of letters by weaving shorter ones and longer ones.
- Select not to overlap statements with similar contents and points.
- Select uniformly each category.

Subjects may be tired before finishing all of the evaluations. If so, reliability decreases. Therefore, twelve sets are divided into three groups, each of which contains four categories. For each group, answer-filing forms are prepared as a book. We have three books: Book A, B and C. Subjects evaluate the impression of the statements in the order of Book A, B and C. For each book, subjects evaluate the impression of the statements in the following order:

1) Book A : §1 < Q1→A1-1→A1-2→A1-3→A1-4 >
→§4 < Q4→A4-1→A4-2→A4-3→A4-4>
→§7 < Q7→A7-1→A7-2→A7-3→A7-4>
→§10 < Q10→A10-1→A10-2→A10-3→A10-4>

2) Book B : §2 < Q2→A2-1→A2-2→A2-3→A2-4>
→§5 < Q5→A5-1→A5-2→A5-3→A5-4>
→§8 < Q8→A8-1→A8-2→A8-3→A8-4>
→§11 < Q11→A11-1→A11-2→A11-3→A11-4>
3) Book C :§3 < Q3→A3-1→A3-2→A3-3→A3-4>
→§6 < Q6→A6-1→A6-2→A6-3→A6-4>
→§9 < Q9→A9-1→A9-2→A9-3→A9-4>
→§12 < Q12→A12-1→A12-2→A12-3→A12-4>

Here, §1 - §3, §4 -§6, §7 -§9 and §10 -§12 are the questions and answers of the categories of Yahoo! auction, PC & peripherals, love counseling & human relationships and political & social problems, respectively. Q indicates a question in the section, while A does an answer to it. For example, Q1 is a question of §1, and A1-1, A1-2, A1-3 and A1-4 are answers to Q1. We asked subjects to read answers after a question because some answers are obscure before a question. The answers Ai-1 (i=1, 2, … , 12) are the "Best Answers" actually chosen at Yahoo! Chiebukuro. This fact, however, is not informed to the subjects.

4.2 Experimental Result

In order to inspect whether or not gender difference affected impressive evaluation, we tested with the significance level at 1% between answers of thirty-three males and those of eight females. This result is shown in Table 2. As a result, there is a significant difference between males and females. Therefore, the scores evaluated by males are used.

Next, in order to measure fatigue effect, we tested with the significance level at 1% among 3 books: Book A, B and C. This result is shown in Table 3. Since the value p is larger than significance level, there is no significant difference among books. Therefore, no fatigue effect can be observed. Impressive evaluation scores of all the books could be used.

We applied factor analysis to the impressive evaluation scores with Varimax rotation. The factors which exist in a set of objects or concepts are obtained by applying the factor analysis. Please see Appendix for further explanation of the factor analysis. By adopting the criterion that eignevalues of their factors are over 1.0, nine factors are obtained. Eigenvalue, contribution ratio and cumulative contribution ratio of each factor are listed in Table 4. Factor burdens are shown in Table 5.

Table 2 Result of Inspecting Significance of Gender

Gender	Average	Variance	Value p
Male	1.255	1.143	1.768E-33
Female	1.627	1.937	

Table 3 Result of Inspecting Significance of Books

Book	Average	Variance	Value p
A	1.228	1.563	
B	1.240	1.660	0.429
C	1.238	1.623	

Table 4 Eigenvalues, Contribution Ratio and Cumulative Contribution Ratio of Each Factor

Factor	Eigenvalue	Contrbution Ratio[%]	Cumulative Contrbution Ratio[%]
1	11.098	14.5	14.5
2	6.777	11.4	25.8
3	2.686	6.1	31.9
4	2.238	3.7	35.6
5	1.575	3.6	39.2
6	1.500	3.5	42.8
7	1.429	3.1	45.9
8	1.184	2.1	48.0
9	1.098	2.0	50.0

The factor burdens, whose absolute values are over 0.5, are shown shaded. The impressive words whose absolute values of factor burdens are large explain the factor.

Interpretation against each factor is as follows:

- 1st factor: Positive evaluation sentences, such as "persuasive," "wonderful," "fulfilling," etc. This factor is called "accuracy."
- 2nd factor: Negative evaluation sentences, such as "uncomfortable," "resentful," "disillusioning," etc. This factor is called "displeasure."
- 3rd factor: Unusual viewpoints and contents of sentences, such as "special," "creative," "original," "unexpected," etc. This factor is called "creativity."
- 4th factor: Clarity of statements, such as "easy" and "clear" as positive, and "difficult," as negative. This factor is called "ease."
- 5th factor: Persistent and detailed statements, such as "minute," "persistent" and "long." This factor is called "persistence."
- 6th factor: Ambiguity of sentences, such as "insufficient" and "ambiguous." This factor is called "ambiguity."
- 7th factor: Moving sentences, such as "impressive" and "warm-hearted." This factor is called "moving."
- 8th factor: Hardship and effort for a long time, such as "touching." This factor is called "effort."
- 9th factor: Enthusiasm and zeal from sentences, such as "hot" and "powerful." This factor is called "hotness."

From the factors obtained, factor scores are calculated and used to estimate the "Best Answers" of any question and answer statements.

5 Factor Scores of "Best Answers"

Here, the factor scores of a question are compared with those of its "Best Answer." Those of §3, which is of Yahoo! auction, are shown in Table 6. The tendency of the factor scores of the question Q3 is quite similar to that of the "Best Answer" (A3-1). The Euclidean distances between Q3 and A3-i (i=1, …. , 4) are also shown in Table 6. The distance between Q3 and A3-1 is shortest among the

Table 5 Factor Burdens of Nine Factors

Impressive Word	1	2	3	4	5	6	7	8	9
Persuasive	0.751	−0.171	−0.114	0.009	0.054	−0.148	0.002	0.065	0.024
Wonderful	0.735	−0.165	0.040	0.061	−0.012	−0.064	0.313	−0.002	0.144
Important	0.722	−0.120	−0.020	−0.090	0.046	0.016	0.012	0.096	0.079
Fulfilling	0.705	−0.122	0.009	−0.037	0.157	−0.063	0.174	−0.015	0.047
Accurate	0.702	−0.220	−0.132	0.114	0.013	−0.201	−0.079	0.014	0.006
Real	0.662	−0.154	−0.165	−0.032	0.106	−0.085	−0.084	0.179	−0.074
Appropriate	0.612	−0.179	−0.166	0.147	0.074	−0.046	−0.067	0.047	−0.012
Skillful	0.588	−0.051	0.090	0.030	0.096	−0.064	0.103	0.025	0.045
Favorable	0.582	−0.384	−0.051	0.109	0.006	−0.123	0.391	0.022	−0.028
Beautiful	0.559	−0.094	0.005	0.027	0.030	−0.007	0.463	0.087	0.216
Courteous	0.542	−0.337	−0.111	0.002	0.189	−0.206	0.136	0.069	−0.081
Fluent	0.511	0.010	0.146	0.175	0.185	0.004	0.071	0.056	0.116
Refreshing	0.510	−0.032	0.205	0.165	−0.129	−0.006	0.210	−0.001	0.138
Uncomfortable	−0.233	0.834	0.079	−0.039	0.124	0.078	−0.025	−0.071	0.004
Resentful	−0.054	0.778	0.103	−0.074	0.074	0.002	0.028	0.093	−0.039
Thoughtless	−0.243	0.728	0.230	0.002	0.070	0.103	0.040	−0.040	−0.001
Disillusioning	−0.212	0.709	0.160	0.032	0.124	0.167	−0.034	0.063	−0.024
Unjust	−0.108	0.658	0.241	−0.091	0.061	0.168	0.047	0.101	−0.051
Regrettable	−0.327	0.627	0.132	0.025	0.129	0.203	−0.062	0.142	−0.051
Amazing	−0.394	0.592	0.177	0.043	0.121	0.261	−0.033	0.033	0.036
Fearful	−0.050	0.528	0.185	−0.073	0.146	0.036	−0.121	0.163	0.177
Creative	0.036	0.087	0.759	0.044	0.055	0.047	0.028	−0.018	0.027
Special	−0.019	0.218	0.756	−0.019	0.108	0.048	0.008	0.039	0.051
Original	0.028	0.175	0.644	−0.012	−0.078	0.018	−0.003	0.138	0.075
Unexpected	−0.088	0.255	0.608	−0.026	0.048	0.067	0.040	0.073	0.021
Marvelous	−0.079	0.261	0.541	−0.095	0.155	0.180	0.055	0.170	−0.011
Easy	0.235	−0.024	−0.012	0.706	0.015	0.058	0.044	0.022	−0.060
Clear	0.369	−0.073	0.040	0.558	−0.057	−0.160	−0.048	0.016	0.040
Difficult	0.169	0.130	0.171	−0.584	0.241	0.250	−0.084	0.128	0.070
Minute	0.293	0.100	0.003	−0.077	0.627	−0.017	0.016	−0.003	0.001
Persistent	−0.019	0.331	0.095	−0.052	0.521	0.054	−0.020	0.135	0.095
Long	0.243	0.039	0.036	−0.168	0.513	−0.117	0.083	0.027	−0.030
Ambiguous	−0.292	0.195	0.146	−0.095	0.021	0.569	0.029	−0.031	−0.017
Insufficient	−0.210	0.338	0.116	−0.095	−0.048	0.564	0.013	−0.019	−0.092
Warm-Hearted	0.470	−0.199	0.033	0.088	0.056	−0.019	0.580	0.212	0.058
Impressive	0.474	−0.041	0.076	−0.020	−0.010	−0.016	0.509	0.263	0.243
Touching	0.129	0.066	0.152	−0.019	0.103	−0.088	0.125	0.558	0.069
Hot	0.323	0.024	0.094	−0.066	0.155	−0.145	0.254	0.185	0.546
Powerful	0.438	0.185	0.145	0.013	0.018	−0.054	0.096	−0.020	0.514
Exaggerating	−0.115	0.283	0.172	0.046	0.381	0.127	0.000	0.194	0.259
Firm	0.238	0.151	0.061	−0.118	0.265	0.149	−0.076	0.059	0.170
Sharp	0.052	0.487	0.052	−0.015	−0.001	0.046	−0.201	0.048	0.144
Suspicious	−0.205	0.445	0.305	−0.026	0.160	0.266	0.015	−0.003	0.130
Simple	0.030	0.150	0.125	0.477	−0.109	0.336	0.024	0.036	0.053
Fun	0.263	−0.059	0.254	0.174	0.000	0.072	0.331	0.085	0.031
Dear	0.162	0.136	0.111	−0.041	0.046	0.063	0.098	0.399	0.026
Inevitable	0.074	0.240	0.017	0.097	0.187	0.212	−0.134	0.247	0.020
Dull	−0.135	0.366	0.022	0.089	−0.105	0.426	−0.218	−0.120	0.018
Complicating	0.217	0.172	0.224	−0.411	0.330	0.157	−0.105	0.152	−0.020
Faltering	−0.096	0.170	0.076	−0.003	0.271	0.374	0.060	0.216	−0.085

distances. This shows that the impression of the question is very similar to that of its “Best Answer.” The same tendency could be found in the question and answer sets of §6, §7, §9, §10 and §12.

The factor scores of the question and those of its “Best Answer” of §4 are also shown in Table 6. The tendency of those of the question Q4 is not similar to that of the “Best Answer” (A4-1). The Euclidean distance between the question

Table 6 Two Sets of Factor Scores and Distance

Statements	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	Distance
Q3	−0.797	−0.376	0.884	0.233	0.352	−0.202	−0.224	0.656	−0.091	–
A3-1	−0.216	−0.417	0.272	0.358	0.390	−0.378	0.005	0.565	−0.223	1.181
A3-2	−0.608	0.508	0.755	0.601	−0.584	−0.143	0.145	−0.249	−0.355	1.413
A3-3	−0.442	0.800	−0.233	0.228	−0.132	−0.524	−0.226	−0.043	−0.146	1.768
A3-4	−0.391	−0.404	0.358	0.402	−0.290	0.145	0.431	0.055	−0.260	1.215

Statements	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	Distance
Q4	−0.245	−0.454	−0.599	0.349	−0.100	0.571	−0.170	0.116	−0.504	–
A4-1	1.118	−0.546	−0.498	−0.188	0.578	−0.205	0.003	−0.434	−0.424	2.357
A4-2	−0.351	0.479	−0.470	−0.158	−0.864	1.232	−0.207	−0.437	−0.299	2.135
A4-3	0.490	−0.382	−0.243	−0.036	−0.146	0.173	−0.094	−0.283	−0.494	1.814
A4-4	−0.294	0.128	0.041	−0.720	−0.027	0.807	−0.366	−0.466	−0.004	2.059

Q4 and its “Best Answer” A4-1 is largest. We have to use another dissimilarity of factor scores rather than Euclidean distance in order to estimate “Best Answers.” Clarifying the dissimilarity of factor scores is included in future work.

6 Feature Values of Statements

6.1 Feature Values of Statements

For estimating factor scores of question and answer statements through multiple regression analysis, feature values of statements effective in the estimation must be clarified. Here, statistics of statements is used as feature values. Statistics of statements includes the length of a statement, numbers of word classes, and their ratios to all the words. In order to obtain statement lengths and numbers of word classes such as nouns and verbs, morphological analysis is applied to 60 statements.

Considering that words are likely to appear more than once in statements, the number of vocabularies and that of words are uniquely extracted. The number of vocabularies is defined as counting a word “one” even if the same word appears in sentences several times, while that of words means its appearance times. For example, in a sentence “I know I feel happy,” the word “I” appears twice, so for a vocabulary “I,” the number of the word “I” is 2. TTR (Type Token Ratio) is the ratio of vocabulary to words.

It can be also said that, as to Chinese characters, symbols and alphanumeric characters, the proportion of their appearance can affect impression. Therefore, not only their number of appearance, but the length of sentences including them and their percentage content in sentences (The number of concerned kinds of letters out of that of whole the letters) are needed to be considered. For example, “Chinese characters (word)” means the number of Chinese characters in sentences, while “Chinese characters (%)” means the percentage of Chinese characters in them.

“Unknown word” means the word judged as “Unknown” in using Text Seer [7] as a default. Those words were subsequently registered on the dictionary as a noun or symbol, and morphological analysis is applied again.

We use 64 feature values listed in Table 7.

6.2 Consideration of Multicollinearity

In applying multiple regression analysis, it is required that explanatory variables have no correlation each other, and the following conditions must be considered.

Table 7 Feature Values of Sentences

f	Feature Values of Sentences
f1	Letters
f2	Noun(Vocabulary)
f3	Verb(Vocabulary)
f4	Adjective(Vocabulary)
f5	Adverb(Vocabulary)
f6	Pre-Noun Adjectival(Vocabulary)
f7	Conjuction(Vocabulary)
f8	Interjection(Vocabulary)
f9	Auxiliary Verb(Vocabulary)
f10	Postpositional Particle(Vocabulary)
f11	Prefixs
f12	Signs(Vocabulary)
f13	Sentences
f14	Average Length of Sentences(Words)
f15	Average Length of Sentences(Letters)
f16	Hiragana(Word)
f17	Chinese Characters(Word)
f18	Katakana(Word)
f19	Signs(Word)
f20	Full-Size Characters(Word)
f21	Alphanumeric Characters(Word)
f22	Full-Size Alphanumeric Characters(Word)
f23	Half-Size Alphanumeric Characters(Word)
f24	Words
f25	Vocabularies
f26	Unknown Words
f27	Noun(Word)
f28	Verb(Word)
f29	Adjective(Word)
f30	Adverb(Word)
f31	Pre-Noun Adjectival(Word)
f32	Conjuction(Word)
f33	Interjection(Word)
f34	Auxiliary Verb(Word)
f35	Postpositional Particle(Word)
f36	Hiragana(%)
f37	Chinese Characters(%)
f38	Katakana(%)
f39	Signs(%)
f40	TTR
f41	Full-Size Characters(%)
f42	Alphanumeric Characters(%)
f43	Full-Size Alphanumeric Characters(%)
f44	Half-Size Alphanumeric Characters(%)
f45	Noun(%)
f46	Verb(%)
f47	Adjective(%)
f48	Adverb(%)
f49	Pre-Noun Adjectival(%)
f50	Conjuction(%)
f51	Interjection(%)
f52	Auxiliary Verb(%)
f53	Postpositional Particle(%)
f54	Exclamation Marks
f55	Question Marks
f56	Periods
f57	Commas
f58	Middle Dots
f59	Three Dots
f60	Quotation Marks
f61	Closing Quotation Marks
f62	Parentheses
f63	Closing Parentheses
f64	Slash Characters

1) Selection of the explanatory variable whose correlation coefficient to a dependent variable is high.
2) Exclusion of one of two criterion variables whose correlation coefficient is high.

If the condition 2) is not satisfied, partial regression coefficients cannot be properly calculated. This situation is called “multicollinearity.”

In order to avoid multicollinearity, either of the explanatory variables, whose correlation coefficients are high, is excluded from explanatory variables. As a result, the number of explanatory variables is thirty-seven, which are shaded in Table 7.

7 Analysis Result

7.1 Considering Only Monadic Terms

Nine factor scores are denoted as $y_1, y_2, \cdots, y_9$. For sixty question and answer statements, multiple regression analysis is applied to the feature values of statements for estimating the factor scores.

As a result, a set of regression expressions (1) is obtained. The expression of 9th factor, however, was unable to be obtained.

Multiple correlation coefficients between the factor scores and those estimated with monadic terms are shown in Table 8. These coefficients are not high.

7.2 Considering Quadratic Terms

We use multiple regression analysis, considering quadratic terms (the product of explanatory variables). For sixty question and answer statements, multiple regression analysis is applied to the feature values of statements for estimating.

As a result, a set of regression expressions (2) is obtained. The expression of 9th factor was unable to be obtained.

Multiple correlation coefficients between the factor scores and those estimated with quadratic terms are also shown in Table 8.

$$\left\{\begin{array}{l} y_1 = 0.00579 f_1 - 0.131 f_{22} + 0.0851 f_{29} + 0.484 f_{62} + 0.0526 f_{43} - 0.0147 f_{38} \\ \quad + 0.121 f_{60} + 0.0101 f_{45} - 0.0740 f_{57} - 0.00228 f_{15} - 0.582 \\ y_2 = -0.0938 f_9 + 0.369 \\ y_3 = -0.0850 f_9 + 0.0444 f_{22} + 0.245 \\ y_4 = -0.00348 f_1 + 0.0588 f_{22} - 0.140 f_{55} + 0.0673 f_9 + 0.181 \\ y_5 = 0.0898 f_9 + 0.0974 f_{55} + 0.0157 f_{18} + 0.0130 f_{37} + 0.403 f_{59} + 0.00304 f_{15} \\ \quad + 0.0268 f_{13} - 1.14 \\ y_6 = -0.00340 f_1 - 0.00828 f_{36} + 0.926 \\ y_7 = 0.456 f_{33} + 0.0836 f_{30} + 0.102 f_{47} - 0.0330 f_{56} - 0.193 \\ y_8 = 0.108 f_9 - 0.00628 f_{42} - 0.00120 f_1 + 0.0826 f_{47} - 0.305 \end{array}\right\} \quad (1)$$

Table 8 Multiple Correlation Coefficient of Each Factor

Factor	Multiple Correlation Coefficient	
	Monadic Terms	Quadratic Terms
1st (accuracy)	0.879	0.873
2nd (displeasure)	0.350	0.703
3rd (creativity)	0.475	0.727
4th (ease)	0.643	0.784
5th (persistence)	0.881	0.864
6th (ambiguity)	0.677	0.690
7th (moving)	0.562	0.575
8th (effort)	0.587	0.720

$$
\left\{
\begin{aligned}
y_1 &= 0.00536 f_{13} f_{15} - 0.532 f_{62} f_{64} + 0.00843 f_{42} + 0.0499 f_{30} f_{47} - 0.00162 f_{15} f_{31} \\
&\quad - 0.00272 f_{13} f_{57} + 0.00958 f_{29} f_{38} - 0.113 f_{48} f_{55} - 0.00807 f_{30} f_{39} + 0.281 f_{50} f_{55} \\
&\quad - 0.721 \\
y_2 &= -0.00282 f_9 f_{36} + 0.0249 f_{43} f_{57} - 0.0141 f_{18} f_{55} - 1.42 f_{40} f_{40} - 0.00630 f_{45} f_{47} \\
&\quad + 0.0195 f_{37} f_{40} + 1.10 \\
y_3 &= 0.579 f_{62} f_{64} + 1.66 f_{40} f_{40} - 0.0139 f_{36} f_{40} + 0.00263 f_{37} f_{43} + 0.0125 f_{18} f_{31} \\
&\quad + 0.00490 f_{56} f_{57} - 0.516 \\
y_4 &= -0.00345 f_{13} f_{15} - 0.214 f_{37} f_{40} - 0.371 f_{48} f_{51} + 0.00229 f_{12} f_{37} + 0.00177 f_{38} f_{39} \\
&\quad + 0.000970 f_{38} f_{42} + 0.543 \\
y_5 &= 0.00509 f_{13} f_{15} + 0.0693 f_{48} f_{55} - 0.341 f_{62} f_{64} - 0.199 f_{43} f_{50} + 0.923 f_{49} f_{59} \\
&\quad - 0.0149 f_{43} - 0.00625 f_{20} f_{55} - 0.554 \\
y_6 &= 1.21 f_{40} f_{40} + 0.0165 f_{37} f_{40} - 0.857 \\
y_7 &= 0.432 f_{33} + 0.0821 f_{30} - 0.00492 f_{56} f_{57} + 0.0982 f_{47} - 0.216 \\
y_8 &= 0.00138 f_9 f_{36} - 0.00442 f_{42} f_{57} - 0.0173 f_{57} f_{60} + 0.00997 f_{18} f_{55} - 0.0824 f_{29} \\
&\quad + 0.303 f_{32} f_{59} + 0.00471 f_{31} f_{45} - 0.177
\end{aligned}
\right\} \quad (2)
$$

7.3 Comparison and Considerations

It is said that when the value of multiple correlation coefficient is over 0.9, the precision of analysis is very good; when the coefficient is more than 0.7, the precision is fairly good; and the precision is poor when the value is below 0.7.

From the result of multiple regression analysis with only monadic terms, the first factor (accuracy) and the fifth factor (persistence) are well estimated because the values of multiple correlation coefficients are over 0.7. The other seven factors, on the other hand, are not estimated well because the values are below 0.7.

From the result of multiple regression analysis with quadratic terms, the first factor (accuracy), the second factor (displeasure), the third factor (creativity), the fourth factor (ease), the fifth factor (persistence) and the eighth factor (effort) are well estimated because the values of multiple correlation coefficients of these factors are over 0.7. The other three factors, on the other hand, are not estimated well because the values are below 0.7. Compared with monadic terms, considering quadratic terms improves multiple correlation coefficient of each factor. The precisions of almost all the factors except the first factor and the fifth factor are improved.

The feature values appearing in the set of regression expressions (2) are said to be effective in estimating factor scores. The average length of sentences (f15) and TTR (Type Token Ratio, f40) are included in more than one expression shown in (2). These feature values may be said to be important in describing the features of question and answer statements.

8 Conclusion

This paper evaluated the impression of the statements of a Q&A site in order to match a questioner and respondents properly. First, we selected fifty impressive

words. With these fifty impressive words, the impressions of the statements are evaluated. Sixty statements of questions and answers really posted at Yahoo! Chiebukuro are used. As a result of factor analysis, nine factors on statements were obtained. By using the factor scores, the correspondence between a question and its "Best Answer" could be calculated.

In order to obtain the factor scores of any other statements, multiple regression analysis is applied to the feature values of statements. Word classes such as noun and verb, and the number of appearance or the percentage of Chinese and alphanumeric characters, which are obtained by morphological analysis, are adopted as the feature values of statements. Considering quadratic terms could improve the precision of the estimation.

The major contributions of this paper are as follows:

- Factor scores of accuracy and persistence can be well estimated by using the regression formulas shown in (2). These two types of factor scores of the statements that are not used for the experiment can be calculated.
- Consideration of quadratic terms improves the accuracy of multiple regression analysis.
- The average length of sentences and TTR (Type Token Ratio) may describe the features of question and answer statements well.

Precision of estimation of some factors, however, are required to be improved. The ninth factor has not been estimated yet. Feature values of statements must be needed to be added so that its multiple regression expression can be obtained. There are some data that are widely known as greatly affecting the language process of people, which seem to be very useful as feature values of sentences [5]. Adopting them as the feature values is included in future work. The experiment with more subjects is also included in future work. Balancing male and female subjects would like to be considered. Finally, building a system introducing a question to the users who could precisely answer the question is included as well in future work.

Acknowledgments. This research is partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (C), 21500091, 2009-2012. This research uses the data of "Yahoo! Chiebukuro" given to National Institute of Informatics by Yahoo Japan Corporation.

References

[1] Yahoo! Answers, `http://answers.yahoo.com/`
[2] Yahoo! Chiebukuro (in Japanese), `http://chiebukuro.yahoo.co.jp/`
[3] Nishihara, Y., Matsumura, N., Yachida, H.: Understanding of Writing Style Patterns between Q&A in Knowledge Sharing Community. In: The 22nd Annual Conference of the Japanese Society for Artificial Intelligence (2008) (in Japanese)
[4] Kumamoto, T.: Creating an Impression Space to Represent Impressions of News Articles. The Institute of Electronics, Information and Communication Engineers, WI2-2008-35, pp.47–52 (July 18, 2008) (in Japanese)
[5] Baayen, R.H.: Nihongo-no goitokusei: Lexical properties of Japanese (Word Imageability). NTT database, vol. 3. Sanseido, Baayen, R. H, Tokyo, Japan (2008) (in Japanese)

[6] Gregory, R.J.: Psychological Testing: History, Principles, and Applications, Allyn & Bacon, Inc. (2000)
[7] Text Seer Manual (in Japanese), http://www.valdes.titech.ac.jp/~t_kawa/ts/manual.html
[8] Blooma, M.J., Chua, A.Y.K., Goh, D.H.L.: A Predictive Framework for Retrieving the Best Answer. In: Proc. of 2008 ACM Symposium on Applied Computing (SAC 2008), pp. 1107–1111 (2008)
[9] Agichtein, E., Castillo, C., Donato, D., Gionis, A., Mishne, G.: Finding High-Quality Content in Social Media. In: Proc. of the Int'l Conf. on Web Search and Web Data Mining (WSDM 2008), pp.183–194 (2008)
[10] Wang, X.J., Tu, X., Feng, D., Zhang, L.: Ranking Community Answers by Modeling Question-Answer Relationships via Analogical Reasoning. In: Proc. of 32nd Int'l ACM SIGIR Conf., pp.179–186 (2009)
[11] Kim, S., Oh, J. S., Oh, S.: Best-Answer Selection Criteria in a Social Q&A site from the User-Oriented Relevance Perspective. In: Proc. of American Society for Information Science and Technology (ASIS&T) 2007 Annual Meeting (2007)
[12] Adamic, L.A., Zhang, J., Bakshy, E., Ackerman, M.S.: Knowledge Sharing and Yahoo Answers: Everyone Knows Something. In: Proc. of 17th Int'l Conf. on World Wide Web, WWW 2008 (2008)
[13] Jurczyk, P., Agichtein, E.: Discovering Authorities in Question Answer Communities by Using Link Analysis. In: Proc. of 16th ACM Conf. on Inf. and Know. Management (CIKM 2007), pp.919–922 (2007)
[14] Hovy, E., Gerber, L., Hermjakob, U., Junk, M., Lin, C.-Y.: Question Answering in Webclopedia. In: Proc. of 9th Text Retrieval Conf., pp. 655–664 (2000)

Appendix

In factor analysis, an $n \times p$ matrix Z is modeled by using the following equation:

$$Z=FA'+E,$$

where F is an $n \times m$ matrix, A' is the transposed matrix of a $p \times m$ matrix A, and E is an $n \times p$ matrix. In order to obtain the matrices F and A, m is selected to become as small as possible, and E is decided to become sufficiently small. The matrix A is called the factor burden matrix. The matrix F is called the factor score matrix. The matrix E is called the error matrix.

Several methods are proposed to make the matrix E sufficiently small. The main factor analysis method makes the sum of the squares of the elements of U minimum, where U is a variance and co-variance matrix of E $(U = E'E)$. The Minres method makes the sum of the squares of the non-singular elements of U minimum.

The factor burden matrix can not uniquely be decided because this matrix has the freedom on the rotation. The matrix is often rotated in order that the absolute values of some factors of this matrix are large, and those of the other ones are small. This is because the rotated matrix can make the explanation of the phenomena easy. The Varimax rotation is one of the most popular methods of this type of rotation.

The factors which exist in a set of objects or concepts are obtained by applying the factor analysis [6].

An Integrated Hierarchical Temporal Memory Network for Continuous Multi-Interval Prediction of Stock Price Trends

Hyun-Syug Kang and Jianhua Diao

Abstract. We propose an integrated hierarchical temporal memory (IHTM) network for continuous multi-interval prediction (CMIP) based on the hierarchical temporal memory (HTM) theory. The IHTM network is constructed by introducing three kinds of new modules to the original HTM network. One is Zeta1FirstNode which is used to cooperate with the original HTM node types for predicting stock price with multi-interval at any given time. The second is ShiftVectorFileSensor module used for inputting stock price data to the network continuously. The third is a MultipleOutputEffector module which produces multiple prediction results with different intervals simultaneously. With these three new modules, the IHTM network make sure newly arriving data is processed and continuous multi-interval prediction is provided. Performance evaluation shows that the IHTM is efficient in the memory and time consumption compared with the original HTM network in CMIP.

Keywords: Hierarchical temporal memory (HTM), Continuous multi-interval prediction (CMIP), Stock price trends.

Hyun-Syug Kang
Department of Computer Science, Gyeongsang National University, Korea
e-mail : hskang@gnu.ac.kr

Jianhua Diao
School of Software, Dalian University of Foreign Languages, China
e-mail : software@dlufl.edu.cn

R. Lee (Ed.): Software and Network Engineering, SCI 413, pp. 15–27.
springerlink.com

1 Introduction

The stock market is a complex and dynamic system with noisy, unstable and chaotic data series of stock price. Due to the complexity and uncertainty of its moving trend, stock price prediction becomes one of the most challenging problems [1, 2]. More and more researchers try to build systems for predicting the trends of stock price effectively. The prediction results of these systems reflect the changing trends of stock price based on a given single fixed time interval which may not satisfy various requirements of investment decisions. For instance, a bullish stock in the market rose steadily in recent days, but within recent hours it fluctuated strongly. Prediction results based on different time intervals may provide help for investors on making flexible investment decisions on a certain stock, for example, whether invest this stock for a long-term or a short-term. For our best knowledge, there has no research being applied to multi-interval prediction for stock market. Under these investigations and analysis, we proposed a new method to provide a continuous multi-interval prediction (CMIP) in this paper.

In order to develop a "truly" intelligent system for CMIP of stock price, we propose an integrated hierarchical temporal memory (IHTM) network based on the hierarchical temporal memory (HTM) theory [5, 6] in this paper. The IHTM network is constructed by using three kinds of new modules Zeta1FirstNode, ShiftVectorFileSensor and MultipleOutputEffector. Among these three new modules, the Zeta1FirstNode is the basic component for modeling and learning, which integrates the properties of original HTM node types Zeta1Node and Zeta1TopNode. The ShiftVectorFileSensor is a data input sensor used in the prediction stage. The IHTM network constructed with these three new modules is efficient in the memory and time consumption compared with the original HTM network in CMIP.

This paper is organized in 7 chapters. Chapter 2 is the related study of HTM for prediction of stock price trends. In chapter 3, we give a brief introduction to the basic concepts in the CMIP and introduce the problem of CMIP with HTM. We present the insights for constructing an IHTM network using three new modules to support CMIP in chapter 4. Chapter 5 presents the structure of IHTM network. In chapter 6, we show the performance evaluation. Finally, we conclude the paper in chapter 7.

2 HTM for Prediction of Stock Price Trends

Hierarchical temporal memory (HTM) is a machine learning model that replicates the structural and algorithmic properties of the neocortex [3].

An HTM network is a collection of linked nodes, organized in a tree shaped hierarchy. A typical HTM network consists of several levels of Zeta1 nodes. Zeta1 nodes have two node types: Zeta1Node and Zeta1TopNode. The Zeta1Node is

composed of a spatial pooler and a temporal pooler, which is used in the lower level of the network for unsupervised learning. The outputs of Zeta1Nodes are propagated to the upper level still to the Zeta1TopNode at the top level. Zeta1TopNode is composed of a spatial pooler and a supervised mapper, which performs supervised learning and generates prediction results based on the category information. The HTM network is restricted to process data of a single prediction result as there is only one Zeta1TopNode.

The HTM network is suitable for predicting the trends of stock price data because it is known to be suitable for modeling and learning the spatial and temporal relationships between features of data in an intelligent mode [4]. Fig. 1 shows a simple three level HTM network used to predict the stock price trends based on a fixed interval. In Fig. 1, input to the network is a stock price data of 12-minute interval with time granularity as 1 minute. Each Zeta1Node at the bottom level receives input from the sensor. The input of each Zeta1Node at the bottom level corresponds to one 1-minute interval. The input of each Zeta1Node at level 2 is the combination of results from the four child nodes which correspond to four 1-minute intervals, namely 4-minute interval. The Zeta1TopNode at the top level covers the entire twelve 1-minute intervals, namely 12-minute interval. The Zeta1TopNode in the example shown in Fig. 1 performs supervised learning based on four trend categories: Trend 0 is a steady upward movement; Trend 1 is a strongly downward movement; Trend 2 is a recovery movement; Trend 3 is a highly volatile movement.

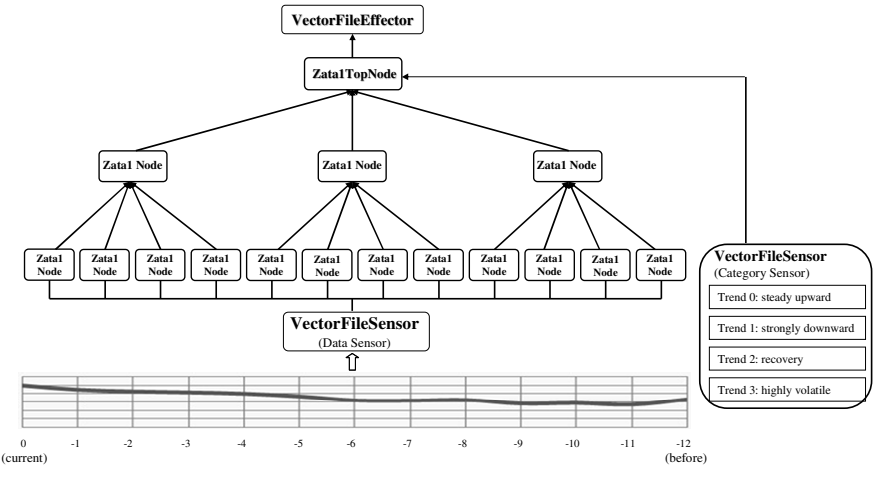


Fig. 1 An example of HTM network to predict stock price trends with a fixed interval

As mentioned above, the HTM network is the same as traditional prediction methods which are restricted to single fixed interval prediction. In order to satisfy diverse requirements of investors, we propose a new prediction concept - continuous multi-interval prediction (CMIP).

3 Continuous Multi-Interval Prediction

In this chapter, we give a brief introduction to the basic concepts in the continuous multi-interval prediction (CMIP) and introduce the problem of CMIP with HTM.

3.1 Concept of CMIP

A series of stock price data can be regarded as a sequence of attribute value vectors $s(t)$ [1]. The vector items based on different time granularity form a continuous flow of data. It maps each time point to a numeric value [6]. Each stock price data has an associated time interval $s(t)$: $t \in [t_e, t_s]$, where t_s is the starting time and t_e is the ending time. The $[t_0, t_{-12}]$ represents the sequence of vectors starting from t_{-12} and ending at t_0.

Prediction for a stock price data is to decide the trends of the changing movement based on its past behaviors. Traditional prediction methods are based on a single fixed length of the time interval of historical data. CMIP is to predict the stock price trends based on various intervals of historical data without interruption. Fig. 2 shows examples of traditional prediction method and CMIP for stock price.

In Fig. 2(a), the left side shows a diagram for a stock price data $s(t)$: $t \in [t_5, t_0]$, where t_0 is the starting time and t_5 is the ending time (current time). For an example, traditional prediction methods can only provide a single prediction result - upward moment at current time t_5 of a fixed time interval from t_5 to t_0. At time t_6, a new value arrives, a single prediction result - upward moment of a fixed time interval from t_6 to t_1 is generated as shown in the right side of Fig. 2(a).

Also take the stock price data shown in the left side of Fig. 2(b) for example, CMIP can provide three prediction results - upward, volatile, upward moment at current time t_5 of three time intervals from t_5 to t_4, t_5 to t_2, and t_5 to t_0 simultaneously as shown in the left side of Fig. 2(b). In this example, a comprehensive prediction result - upward moment obtained by using these three prediction results. As shown in the right side of Fig. 2(b), CMIP can provide three prediction results - downward, downward, volatile moment at current time t_6 of three time intervals from t_6 to t_5, t_6 to t_3, and t_6 to t_1 simultaneously. Likewise, a comprehensive prediction result - downward moment obtained by using these three prediction results.

Therefore, we believe CMIP is able to reduce investment risks in the stock market by comprehensive multiple intervals prediction results.

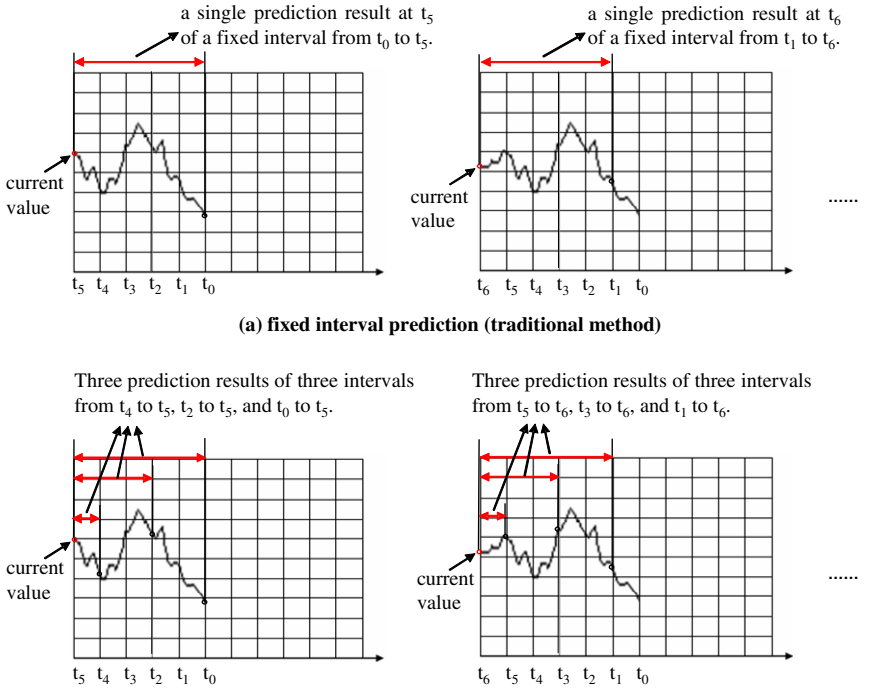


Fig. 2 Examples of traditional prediction method and CMIP for stock price

3.2 Problem of CMIP with HTM

As depicted in chapter 2, the HTM network is restricted to predict the trends of stock price with a single fixed interval. In order to support continuous multi-interval prediction, we have to make multiple HTM networks based on different intervals. Fig. 3 shows an example of three HTM networks based on the three different fixed intervals.

In Fig. 3, network 1 processes stock price data with 1-minute interval. Network 2 processes stock price data with 4-minute interval. Network 3 processes stock price data with 12-minute interval. The number of networks is proportional to the number of prediction intervals and the size of network is proportional to the length of the interval. Obviously, it is not easy to operate and manage the networks. Moreover, it will consume more memory and computation time.

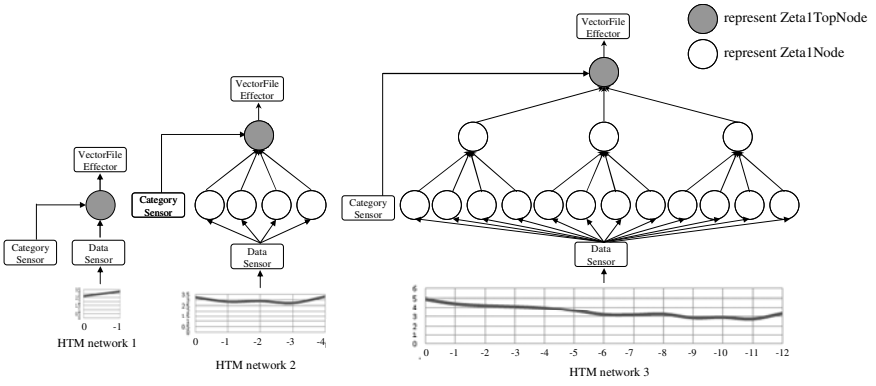


Fig. 3 Multiple HTM networks for the multi-interval prediction

4 IHTM Network Using Three New Modules

To solve the CMIP with HTM problem, a method which integrates multiple HTM networks is needed. In this chapter, we propose an integrated hierarchical temporal memory (IHTM) network using three kinds of new modules.

4.1 Three Kinds of New Modules

We propose three kinds of new modules Zeta1FirstNode, ShiftVectorFileSensor and MultipleOutputEffector to construct an IHTM network.

4.1.1 Zeta1FirstNode

As mentioned in chapter 2, Zeta1Node is used in the lower level for unsupervised learning in the original HTM network. The outputs of Zeta1Nodes are propagated to the upper level till the Zeta1TopNode at the top level. Zeta1TopNode performs supervised learning and generates prediction results based on the category information. It means the original HTM network is restricted to process a single fixed interval as there is only one Zeta1TopNode.

In order to process multi-interval data and generate multiple prediction results, we propose a new node type, namely Zeta1FirstNode. Zeta1FirstNode is the integration of the original HTM nodes Zeta1Node and Zeta1TopNode. As shown in Fig. 4, the Zeta1FirstNode is composed of a spatial pooler, a temporal pooler and a supervised mapper. That is, the Zeta1FirstNode combines the functions of both Zeta1Node and Zeta1TopNode. Thus it can perform both unsupervised and supervised learning. The spatial pooler, temporal pooler, and supervised mapper of Zeta1FirstNode are nearly identical to the three modules of original HTM network.

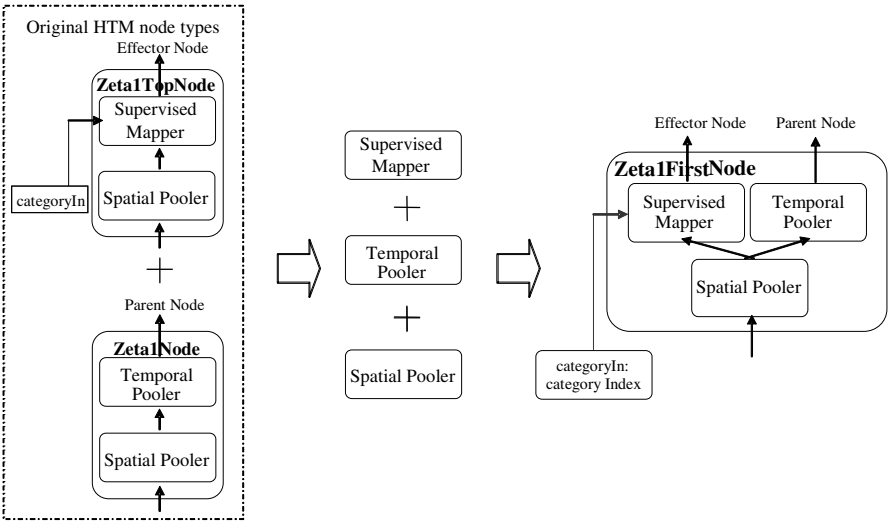


Fig. 4 Structure of Zeta1FirstNode

4.1.2 ShiftVectorFileSensor

In order to process the continuously inputted data to the network and generate results in continuous form, we propose a new data input sensor - ShiftVectorFileSensor to deal with the input sequence of stock price data in prediction stage.

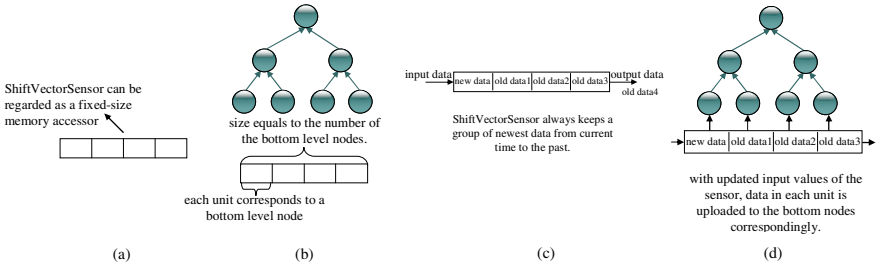


Fig. 5 ShiftVectorFileSensor

As shown in Fig. 5(a), the ShiftVectorFileSensor can be regarded as a single, fixed-size memory accessor. Its size equals to the number of bottom level nodes, each unit of the ShiftVectorFileSensor corresponds to a bottom level nodes. In the example shown in Fig. 5(b), the size of the sensor is 4 because the number of bottom level nodes of the example network is 4. Every time when new data of arrives, it is simply inputted from the left of the sensor stored in a single unit of the sensor, all the existing data in the sensor are right shifted concurrently and the data

in the rightmost unit is overflowed and ignored. In the example shown in Fig. 5(c), present input is marked new data which is inputted from the left of the sensor. All the existing data in the sensor marked old data 1, old data 2, old data 3, and old data 4 are right shifted concurrently. The data in the rightmost unit is old data 4 which is overflowed and ignored. The sensor always keeps a group of newest data from current time to the past. With updated input values of the sensor, data in each unit is uploaded to the bottom nodes correspondingly as shown in Fig. 5(d).

4.1.3 MultipleOutputEffector

As mentioned earlier, our IHTM network is able to predict the trends of stock price relied on current stock price and previous stock prices with different intervals. However, without enough and valid data about the previous stock prices, the prediction based on other longer intervals is invalid. For instance, if the network hasn't received the stock price over 4 minutes, it couldn't provide valid prediction with 4-minute or 12-minute interval. In order to solve this problem, we propose a MultipleOutputEffector to generate the multi-interval prediction result of the network and determine whether these outputs are valid.

MultipleOutputEffector contains a vector of tuples of three values. The number of elements of the vector is equal to the level number of the IHTM network. If the level number is n, the n-ary vector, labeled a, contains 3-value tuples $<e_i, v_i, c_i>$ for each VectorFileEffector i as elements:

$$a[n] = [<e_1, v_1, c_1>, <e_2, v_2, c_2>, \dots, <e_i, v_i, c_i>, ..., <e_n, v_n, c_n>] \quad .... \quad (1)$$

In formula (1), i is the index of level and equal to 1, 2, ..., n; v_i is the validation threshold for determine whether the prediction is valid. This validation threshold equals to the interval length of processing data at level i; c_i is a counter. When the MultipleOutputEffector receives data from the VectorFileEffector at different levels, it will increase the counter and check if it has received enough data for valid prediction by comparing the counter with threshold. If the counter is equal to the threshold, that means MultipleOutputEffector has received enough input vectors, then the input is valid and the subsequent input are also valid.

5 IHTM Network for CMIP

The ITHM network can be considered as an integration of multiple HTM networks where the HTM networks are based on different intervals separately. Take the three HTM networks shown in Fig. 3 for example, the IHTM network can achieve the functions same as the three HTM networks, but with lower memory and computation time consumption.

IHTM is organized as a tree-shaped hierarchy using three types of nodes: Zeta1Node, Zeta1FirstNode, and Zeta1TopNode, together with VectorFileSensor,

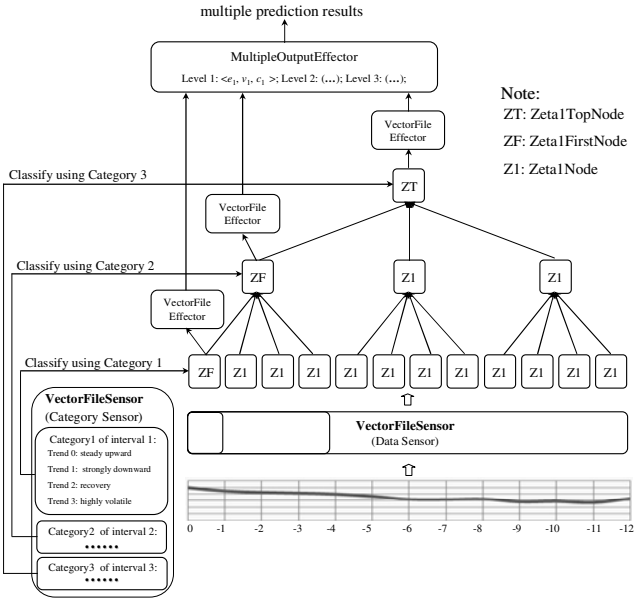


Fig. 6 IHTM network for CMIP

VectorFileEffector, and MultipleOutputEffector. The Zeta1Node, Zeta1FirstNode, and Zeta1TopNode are the basic algorithm and memory modules. The sensor and effector are used for input and output.

Fig. 6 shows an IHTM network with three levels: the bottom level, the middle level, and the top level. There are twelve nodes at the bottom level, where the Zeta1FirstNode is located in the first position, and the remaining eleven locations use the Zeta1Nodes. There are three nodes at the middle level, where the Zeta1FirstNode is also located in the first position, and the remaining two locations use the Zeta1Nodes. The top level has and only has one Zeta1TopNode. In the network, the Zeta1FirstNode and the Zeta1TopNode are used exclusively for supervised training based on the category information from the VectorFileSensor (Category Sensor). The first node at each level receives the newest data firstly, hence the Zeta1FirstNode are used at the first position of each level of the network. The three levels with the sixteen nodes are completely connected to form a hierarchical network.

6 Performance Evaluation

The IHTM network is efficient in reducing memory and time consumption compared with the original HTM network in CMIP. The analysis is based on the original HTM network shown in Fig. 3 and IHTM network shown in Fig. 6.

6.1 Comparison of Memory Consumption

The original HTM network needs multiple networks to achieve CMIP and consumes more memory. The number of original HTM networks increases proportionally with the number of prediction intervals. However, the IHTM network can support CMIP using only one network.

The memory consumption of the original HTM network with n intervals prediction, denoted by Mo, and the IHTM network with n intervals prediction, denoted by Mi, are represented by the following equations:

$$Mo = \sum_{i=1}^{n} mo_i = \sum_{i=1}^{n} (n_{iZ1} \times m_{Z1} + m_{ZT})$$

$$Mi = n_{Z1} \times m_{Z1} + n_{ZF} \times m_{ZF} + m_{ZT}$$

where mo_i is the memory consumption of the ith original HTM network, n_{iZ1} is the number of Zeta1Nodes in the ith HTM network, m_{Z1} is the memory consumption of one Zeta1Node, and m_{ZT} is the memory consumption of one Zeta1TopNode, n_{Z1} is the number of Zeta1Nodes in the IHTM network, n_{ZF} is the number of Zeta1FirstNodes in the IHTM network, and m_{ZF} is the memory consumption of one Zeta1FirstNodes.

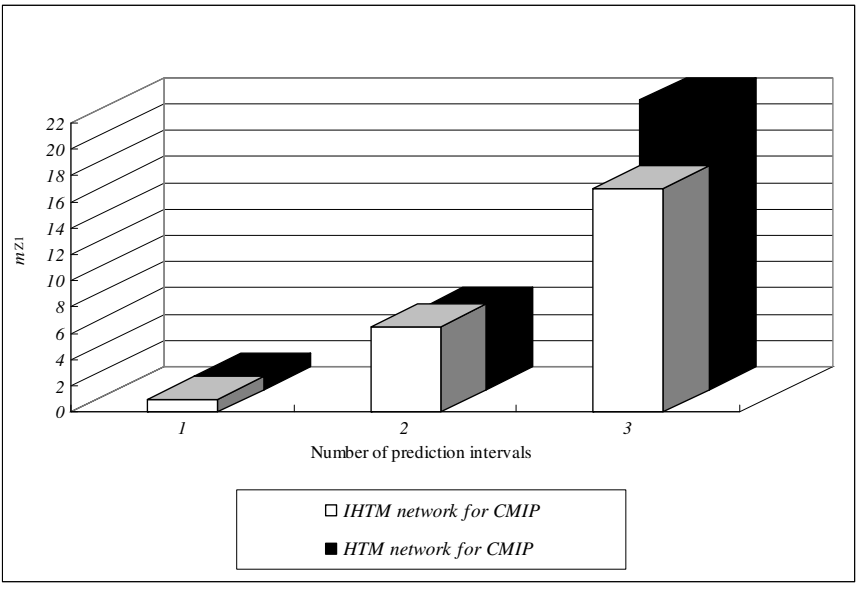


Fig. 7 Comparison of the memory consumption of the original HTM network and the IHTM network

Based on the experimental measurements of these examples, the average memory consumption of one Zeta1Node is nearly equal to that of the Zeta1TopNode. The average memory consumption of one Zeta1FirstNode is almost 1.5 times higher than the Zeta1Node. Using the above equation, let n be 3, we calculate that $mo_1 = n_{1Z1} \times m_{Z1} + m_{ZT} = 0 + m_{ZT} = m_{Z1}$, $mo_2 = n_{2Z1} \times m_{Z1} + m_{ZT} = 4 \times m_{Z1} + m_{ZT} = 5m_{Z1}$, and $mo_3 = n_{3Z1} \times m_{Z1} + m_{ZT} = 15 \times m_{Z1} + m_{ZT} = 16m_{Z1}$. Therefore, the memory consumption of the original HTM network is $Mo = mo_1 + mo_2 + mo_3 = m_{Z1} + 5m_{Z1} + 16m_{Z1} = 22m_{Z1}$. The time consumption of IHTM network is: $Mi = n_{Z1} \times m_{Z1} + n_{ZF} \times m_{ZF} + m_{ZT} = 13 \times m_{Z1} + 2 \times m_{ZF} + m_{ZT} = 13 \times m_{Z1} + 3 \times m_{Z1} + m_{Z1} = 17m_{Z1}$. Fig. 7 shows that the original HTM network consumes much more memory than the IHTM network while the number of prediction intervals increases.

6.2 Comparison of Time Consumption

Besides memory consumption, the time consumption of the IHTM network is much lower than the original HTM network in prediction.

The total time consumption of prediction with the original HTM network and the IHTM network with n intervals are modeled via the following equations:

$$To = \sum_{i=1}^{n} to_i = \sum_{i=1}^{n} (n_{iZ1} \times t_{Z1} + t_{ZT})$$

$$Ti = n_{Z1} \times t_{Z1} + n_{ZF} \times t_{ZF} + t_{ZT}$$

where to_i is the time consumption of the ith original HTM network, n_{iZ1} is the number of Zeta1Nodes in the ith HTM network, t_{Z1} is the time consumption of one Zeta1Node, and t_{ZT} is the time consumption of one Zeta1TopNode; n_{Z1} is the number of Zeta1Nodes in the IHTM network, n_{ZF} is the number of Zeta1FirstNodes in the IHTM network, t_{ZF} is the time consumption of one Zeta1FirstNode.

The average time consumption of prediction of one Zeta1Node is equal to that of the Zeta1TopNode. The average time consumption for prediction of one Zeta1FirstNode is almost 1.4 times higher than the Zeta1Node. Using the above equation, let n be 3, we calculate that $to_1 = t_{ZT} = t_{Z1}$, $to_2 = n_{2Z1} \times t_{Z1} + t_{ZT} = 4 \times t_{Z1} + t_{ZT} = 5\, t_{Z1}$, and $to_3 = n_{3Z1} \times t_{Z1} + t_{ZT} = 15 \times t_{Z1} + t_{ZT} = 16\, t_{Z1}$. Therefore, the time consumption of the original HTM network is $To = to_1 + to_2 + to_3 = t_{Z1} + 5t_{Z1} + 16t_{Z1} = 22t_{Z1}$. The time consumption of the IHTM network is: $Ti = n_{Z1} \times t_{Z1} + n_{ZF} \times t_{ZF} + t_{ZT} = 13 \times t_{Z1} + 2 \times t_{ZF} + t_{ZT} = 13 \times t_{Z1} + 2.8 \times t_{Z1} + t_{Z1} = 16.8t_{Z1}$. As shown in Fig. 8, the result indicates that the IHTM network is significantly better in time consumption than the original HTM network for CMIP with the increase of the number of prediction intervals.

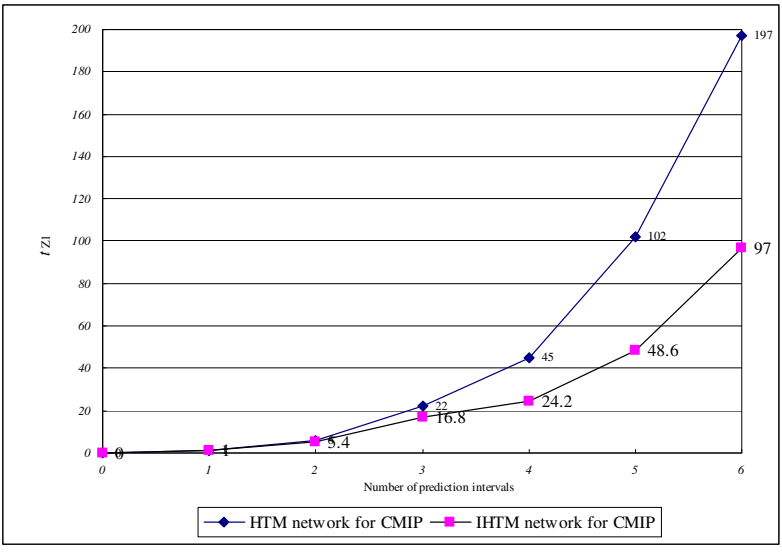


Fig. 8 Comparison of the time consumption of the original HTM network and the IHTM network

7 Conclusion

We present the continuous multi-interval prediction (CMIP) for stock market. The CMIP attempts to predict the trends of stock price based on various intervals of historical data according to different requirements. We believe that CMIP is a crucial problem in the prediction of stock price trends.

We proposed an integrated hierarchical temporal memory (IHTM) network to solve the problem of CMIP for stock market. The IHTM is constructed by introducing three kinds of new modules Zeta1FirstNode, ShiftVectorFileSensor, and the MultipleOutputEffector to the original hierarchical temporal memory (HTM) network. With these three new modules, the IHTM network overcomes the limitation of the original HTM network on the CMIP problem. Performance evaluation shows that the IHTM is efficient in the memory and time consumption compared with the original HTM network in CMIP.

We identified several opportunities for future research. We believe that there is great potential for further research with the prediction of stock price trends using the IHTM network for CMIP. And also, the continuous multi-interval prediction can be applied in various domains, such as analyzing airplane performance and diagnosing diseases amongst others. Future studies are expected to achieve a higher flexibility of the IHTM network on the CMIP problem of stock price and apply the CMIP to the applications which is mentioned above.

Acknowledgments. This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2011-0007857).

References

1. Arel, I., Rose, D., Karrowski, I.: Deep Machine Learning-A New Frontier in Artificial Intelligence Research. IEEE Computational Intelligence Magazine, 13–18 (November 2010)
2. Lee, K., Jo, G.: Expert system for predicting stock market timing using a candlestick chart. Expert system With Applications 16, 357–364 (1999)
3. Hawkins, J., George, D.: Hierarchical Temporal Memory: Concepts, Theory, and Terminology. Numenta Inc. (2006)
4. Numenta, Wallstreet Example. Numenta Inc. (2007), `http://numenta.com/`
5. Numenta, Numenta HTM Home page. Numenta Inc. (2007), `http://numenta.com/`
6. Yang, Y., Wu, X., Zhu, X.: Proactive-reactive Prediction for Data Streams. Data Mining and Knowledge 13, 261–289 (2006)

Web Prefetching by ART1 Neural Network

Wenying Feng, Toufiq Hossain Kazi, and Gongzhu Hu

Abstract. As the Web becomes the major source for information and services, fast access to needed Web objects is a critical requirement for many applications. Various methods have been developed to achieve this goal. Web page prefetching is one of these methods that is commonly used and quite effective reducing the user perceived delays. In this paper, we proposed a new prefetching algorithm, called pART1, based on the original ART1 algorithm that is a neural network approach for clustering. We modified the ART1 algorithm to obtain 2-way weights (bottom-up and top-down) between the clusters of hosts and the URLs (Web pages), and use these weights to make prefetching decisions. In the experiments we conducted, the new algorithm outperformed the original ART1 in terms of cache hit ratio.

Keywords: ART1 neural network, Web object prefetching, Web caching, request pattern vector.

1 Introduction

Web page prefetching is a commonly used approach to improve the response time of page requests from users. The basic idea is to fetch certain Web pages from a remote Web server and store in the user's local cache in advance in anticipation that the user will request pages in the near future that are likely in the local cache so accessing to the remote server may not be needed. The main objective of prefetching algorithms is to maximize the *cache hit ratio*, namely, maximizing the likelihood that the near-future page requests are already in the cache.

Wenying Feng · Toufiq Hossain Kazi
Departments of Computing & Information Systems and Mathematics,
Trent University, Peterborough, Ontario, Canada, K9J 7B8
e-mail: (wfeng,kazihossain)@trentu.ca

Gongzhu Hu
Department of Computer Science, Central Michigan University,
Mt. Pleasant, MI 48859, USA
e-mail: hu1g@cmich.edu

R. Lee (Ed.): Software and Network Engineering, SCI 413, pp. 29–40.
springerlink.com

Various Web page prefetching methods have been proposed and used in applications, such as probability-based and clustering-based approaches. In clustering-based approach, prefetching decisions are made using the information about the *clusters* containing pages that are "close" to each other in the way that if one of them is requested currently, others are likely to be requested in the near future. Similarly, it may also cluster the users according to their Web access patterns that are represented as access vectors. The Adaptive Resonance Theory (ART) neural network is a method that is often used for classification or clustering input patterns. A simplified version, ART1, accepts input vector of binary patterns (with values 0 or 1).

In this paper, we propose a prefetching algorithm, called pART1, that is a revised and improved ART1 application in prefetching [6]. The improvement relies on the method of extraction of the input vectors from the access patterns and the definition of similarity among the vectors. The prefetching decisions are made based on the bottom-up and top-down weights in the network between the clusters of hosts and the URLs. The idea of this new algorithm was introduced in our previous work [9], but that was only a preliminary work without in-depth analysis and performance evaluation of the algorithm nor experiments to valid the analysis. We have since completed the algorithm design, implementation, experiments, and analysis to report in this paper. Experiments were conducted using several data sets obtained by both Monte-Carlo simulations and real-world applications. The results show that pART1 constantly performed somewhat better than the original ART1 algorithm in terms of cache hit ratio with various cache sizes and different access patterns, including an extensive testing using a large data set collected from a digital library system.

2 Basic Architecture of ART1

The basic principles of the Adaptive Resonance Theory (ART) was introduced by Stephen Grossberg in the 1970's [4] as a model for object identification that results from the interaction between multiple layers of sensory information and the processing units in a neural network. This theory was also summarized in a recent encyclopedia [2]. As a class of ART, ART1 is the simplest that accepts input vectors of binary values. To be complete, we give a brief description of ART1 in this section.

ART1 neural network contains three groups: an *input processing unit* (called F_1 layer), a set of *cluster units* (the F_2 layer), and *reset unit* R. The F_1 layer can be divided into two parts: input portion $F_1(a)$ and interface portion $F_1(b)$. Interface layer combines signals from input and F_2 layer. It is used for comparing the similarities between the input signal and weight vector of cluster unit which we select as a candidate of learning. The reset unit R controls the degree of similarity of patterns placed on the same cluster. In addition, two supplemental *gain control units* G_1 and G_2 are included that send and receive signals from all the other units. Excitatory signals are indicated by $+$ and inhibitory signals by $-$. Fig. 1 shows the basic architecture.

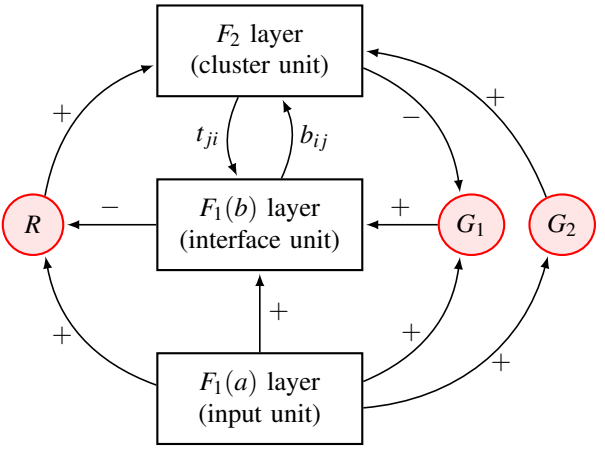


Fig. 1 Basic architecture of ART1 neural network

In Fig. 1, b_{ij} is the bottom-up weight of the connection $X_i \to C_j$ for $X_i \in F_1(b), C_j \in F_2$; and t_{ji} is the top-down weight of the connection $C_j \to X_i$ for $C_j \in F_2, X_i \in F_1(b)$. Each unit in $F_1(a)$ is connected to the corresponding unit in $F_1(b)$. Each unit in $F_1(a)$ and $F_1(b)$ layer is connected to the reset unit R, which in turn is connected to every F_2 unit. F_2 is a competitive layer in which only the uninhibited node with the largest net input has a nonzero activation [3]. The ART1 clustering algorithm makes cluster assignments of the input vectors according to the bottom-up and top-down weights that are iteratively adjusted.

3 pART1 – Prefetching by ART1

To introduce pART1, the new prefetching algorithm based on the ART1 approach, we first give some notations that will be used in the sequel.

n	number of URLs in the input vector;
m	maximum number of clusters;
h	number of hosts;
b_{ij}	bottom-up weights of the connection from $F_1(b)$ to F_2;
t_{ji}	top-down weights of the connection from F_2 to $F_1(b)$;
ρ	vigilance parameter that decides whether the input vector falls into the cluster;
$\boldsymbol{S}$	binary input vector (an n-tuple) in $F_1(a)$ layer;
$\boldsymbol{X}$	activation vector for $F_1(b)$ layer (binary);
$\|\boldsymbol{S}\|$	norm of $\boldsymbol{S}$ defined as $\|\boldsymbol{S}\| = \sum_{i=1}^{n} S_i$;
$\|\boldsymbol{X}\|$	norm of $\boldsymbol{X}$ defined as $\|\boldsymbol{X}\| = \sum_{i=1}^{n} X_i$;

where $\boldsymbol{S} = (S_1, S_2, \cdots S_n)$ and $\boldsymbol{X} = (X_1, X_2, \cdots X_n)$.

3.1 Input Vectors

To apply ART1 in web prefetching, we first need to determine the top-down and bottom-up weights of the network. This is completed by the learning process from the training data. Assume the training data was obtained from h hosts with access to n URLs indexed as URL_j, $j \in \{1,2,\ldots,n\}$, where n and h are positive integers. The input vectors can be represented by a $h \times n$ matrix (a_{ij}):

$$\begin{pmatrix} 1 & 1 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & 1 & \cdots & 0 \\ 0 & 1 & 1 & 0 & \cdots & 1 \\ 0 & 1 & 0 & 1 & \cdots & 0 \\ .. & .. & .. & .. & \cdots & .. \\ 1 & 0 & 1 & 0 & \cdots & 0 \end{pmatrix}$$

Each row of the matrix represents a host and each column represents a URL. The value a_{ij} at cell $[i,j]$ represents the access frequency of host i to URL_j. A value of 1 denotes a higher access frequency and 0 implies a lower access frequency. The important step is how to determine the cutoff, or the threshold access number for the value of 0 and 1. In [10], a_{ij} is assigned 1 if the host i requested the URL_j twice or more then twice. Otherwise, it is assigned 0. However, since the request pattern from each host is different and the threshold 2 may not always be the best for the purpose of future prediction. The method of [10] is useful for small number of requests and very few URLs are requested more than twice. But for bigger system this approach obviously is not a good solution since almost all URLs are requested more than twice. Unnecessary prefetching will increase the cost of the system.

Since different system has different access patterns, we naturally think that a dynamic threshold based on the access pattern would produce a better result. In [9], the average access number of host i to all URL_j is selected as the threshold. This approach has the advantage of automatically assign 0 for URLs with very small number of requested. It is also suitable for the access pattern of a group of URLs are requested more times and others are requested much lower number of times. However, in the case of only a small number of pages have high access and also only a small number of pages have low access, the majority of pages are in the middle, the average URL access number does not count those pages that are accessed just below the average. To overcome this disadvantage, another reasonable approach is to use the ordered access frequency sequence. Let r_{ij} be the request number of host i to URL_j, $i \in \{1,2,\ldots,h\}$ and $j \in \{1,2,\ldots,n\}$. For each host i, sort the sequence $\{r_{i1}, r_{i2}, \ldots, r_{in}\}$ in descending order, we obtain $\{r_1, r_2, \ldots, r_n\}$. Thus $r_k \in \{r_{i1}, r_{i2}, \ldots, r_{in}\}$ and $r_k \geq r_{k+1}$ for $k \in \{1,2,\ldots,n-1\}$. Let $g_k = r_k - r_{k-1}$, g_k is the gap between the access number r_{k-1} and r_k. Let g_m be the maximum gap, thus $g_m = \max\{g_1, g_2, \ldots, g_{n-1}\}$. We select g_{m-1} as the threshold for host i. Thus,

$$a_{ij} = \begin{cases} 1 & \text{if } r_{ij} > g_{m-1}, \\ 0 & \text{otherwise .} \end{cases}$$

The selection of the threshold affects the complexity of the algorithm, training time, number of pages in clusters and rate of successful prediction in prefetching. For a particular system, the best way is to develop a formula based on the analysis of access patterns. The simulation results presented next section were obtained using the average of the access number as the threshold. In addition, different from the traditional ART1 algorithm, the similarity of two binary vectors $\boldsymbol{X} = (x_1, x_2, \ldots, x_n)$ and $\boldsymbol{S} = (s_1, s_2, \ldots, s_n)$ are measured by the norm of difference:

$$\frac{\|\boldsymbol{X} - \boldsymbol{S}\|}{n} \tag{1}$$

where

$$\boldsymbol{X} - \boldsymbol{S} = (d_1, d_2, \ldots, d_n),$$

and

$$d_i = \begin{cases} 1 & \text{if } x_i = s_i, \\ 0 & \text{if } x_i \neq s_i. \end{cases}$$

3.2 Execution of ART1 in Prefetching

As described in Section 2, the basic structure of ART1 neural network has three layers: F_1(a) (input), F_1(b) (interface), F_2 (cluster). In our model, we present input vectors in F_1(a) layer. All the URLs are presented in F_1(b) layer. F_2 layer is the output layer.

After executing the algorithm, two matrices are obtained: one represents the top-down weights and the other represents the bottom-up weights. Bottom-up weight denotes the connections between URLs to clusters. On the other hand, the top-down weight represents the connections between the clusters to URLs. ART1 algorithm clusters each request pattern which is, in our case, the input vector into a cluster. In the process, the number of clusters is an input parameter. For example, if five URLs are divided into three clusters, the bottom-up matrix may look like following:

$$\begin{pmatrix} 0.21 & 0.115 & 0.34 \\ 0.126 & 0.43 & 0.56 \\ 0.37 & 0.52 & 0.29 \\ 0.452 & 0.65 & 0.24 \\ 0.187 & 0.525 & 0.43 \end{pmatrix}$$

An example of the top-down matrix can be given as:

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Here each row in the bottom-up matrix represents URLs and each column represents *clusters*. AS a machine learning algorithm, the top-down and bottom-up weights are

initialized using the training data. When an URL is requested by the user and a cache miss happened, the bottom-up matrix is first used to determine the cluster. For instance, assume that URL_2 is requested and it is not found in the cache. The bottom-up weight connection from URL_2 to clusters is shown in Fig. 2. We can see that $cluster_3$ is connected with the largest bottom-up weight.

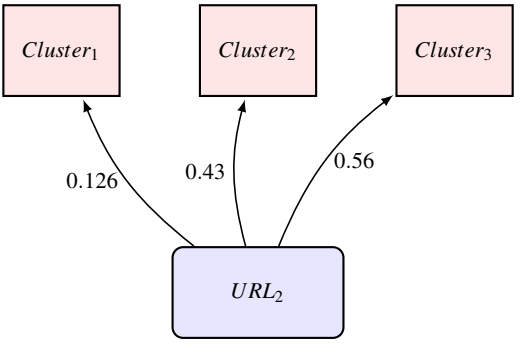


Fig. 2 Bottom-up weight connection

After picking up $cluster_3$, the top-down matrix is used to select URLs to be prefetched. The URLs that are connected with top-down weight 1 are selected. For $cluster_3$, URL_1, URL_2 and URL_3 are connected with top-down weight 1, as illustrated in Fig. 3. As the user requests URL_2, while putting URL_2 into the cache, URL_1 and URL_3 are also moved into the cache.

In our model, prefetching is only conducted when a cache miss happens. In the case of a cache hit, no prefetching is performed considering the increase of network traffic. The pART1 prefetching algorithm is outlined in Algorithm 1, which is a revised version of the one presented in [9].

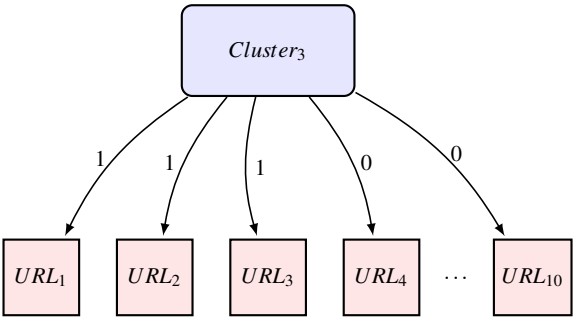


Fig. 3 Top-down weight connection

Algorithm 1: Prefetching Algorithm pART1

Input: u, a requested URL.
Output: U, set of prefetched URLs.

```
1  begin
2      U ← ∅;
3      if u is not in the cache then
4          Find J such that b_iJ = max_j(b_ij) for all node j of URL_i;
5          Find C_k that is connected to u with b_iJ;
6          foreach u_j ∈ URLs connected to C_k do
7              if t_kj = 1 then
8                  if u_j is not in cache then
9                      U ← U ∪ {u_j};
10         end
11     return U;
12 end
```

4 Experimental Results

To evaluate the performance of the new model, a simulation program was developed using Java. The cache is implemented using the commonly applied Least Recent Used [1, 11] replacement policy.

4.1 Monte-Carlo Simulation

To verify and investigate the conditions that ensure our models perform efficiently, we first use input generated from Monte Carlo simulation. It has the advantage of producing access patterns for different page request distributions. For the test, 100,000 requests for 5 hosts with 100 different URLs are generated. To make the testing more realistic, we investigate three different scenarios. First, we assume that a large chunk of URLs are very popular (first 50 URLs with 70% request probability). Second, it is assumed that only a small chunk of URLs are very popular (first 10 URLs with 70% probability). Lastly, it is assumed that there is not much difference in page popularity (first 50 URLs with 50% probability). By varying cache size from $5\% - 50\%$ of total server size, we compare cache hit rate of prefetching using ART1 and the pART1 algorithm. The experimental results for the three cases are shown in Fig.'s 4, 5, and 6, respectively.

Fig. 4 and Fig. 6 show similar trend. When the cache size is small (less than 20%), there is not much differences in hit rates between the two algorithms. However, when cache size is increased, it is noticed that pART1 performs better. The difference between the hit rates increases almost linearly when cache size is increased. The results are expected in the way that a large chunk of URLs

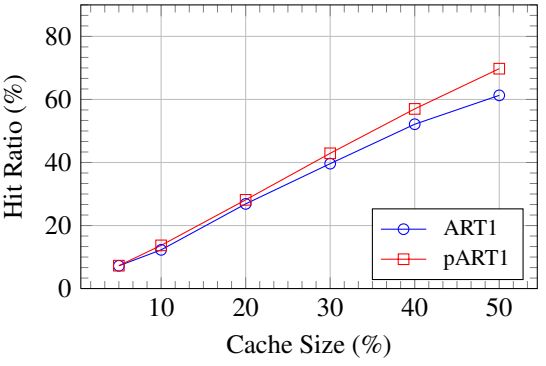


Fig. 4 A large chunk of URLs are very popular

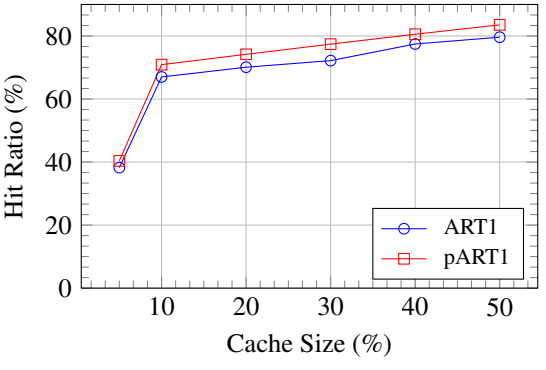


Fig. 5 A small chunk of URLs are very popular

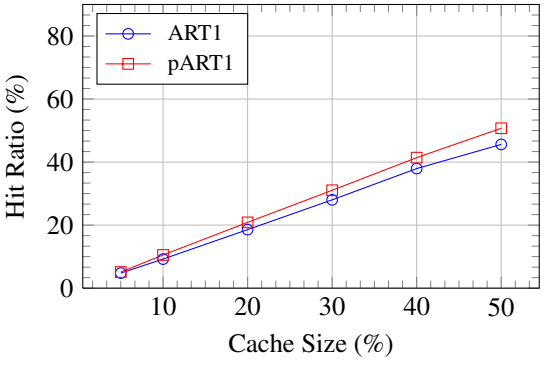


Fig. 6 No pages are popular

are popular should produce similar results as that no pages are popular since we assumed that the request probabilities for each URL of the large chunk are uniform.

Different from Fig.'s 4 and 6, when a small chunk of pages are popular, the hit rates from both ART1 and pART1 are quickly increased for small cache size (less than 10%). When the cache is increased from 10% to 50%, the increase of the hit rate is much slower. This is due to most requests are only for a small group of pages and when the cache size is getting bigger, the small group of pages have always been in the cache and therefore, hit rates are not dramatically increased with cache size.

In all cases, the pART1 algorithm outperformed the original ART algorithm in terms of cache hit ratio, even the improvements are not very significant. It also suggests that pART1 likely performs better in other possible request probabilities.

4.2 Real World Data

The simulation program is also tested using real world data collected from the Trent University digital library. As an important application, digital library has been an active research area. In [8], the authors studied the transaction log analysis of digital library, while in [5], the user access behavior is studied. The effect of caching and prefetching in digital library was investigated in [6] and [7].

Trent library was using text based references on early 90s. At late 90s they moved into web based system. Here all the request comes in library catalog server which is also known as Topcat server. All the documents are stored in catalog server. Catalog server processes user's requests and delivers pages to clients.

The data requested were collected during the period of February 1–28, 2010. There were total 868,412 requests, 406,008 different URLs and requests came from 15,778 different hosts. Each request contains the following fields:

Host_IP	The host address
Rrquest_Time	requested time
URL_Address	The requested URL

From that data, we used 200,000 requests for training (to extract the input vector) and 200,000 data records for testing. Different from the results obtained in Section 4.1, it was found that the pART1 is more efficient for small cache size. For 5% cache size, the hit rate of pART1 is increased 4.77% which is 17% increasing from the ART1 prefetching algorithm. When the cache size is increased, the difference of hit rates between the two algorithms is smaller but again the pART1 is constantly better than the traditional ART1 algorithm. The result is given in Table 1 and shown graphically in Fig. 7.

Table 1 Comparison between traditional ART1 and pART1

Cache size (%)	**Hit ratio** (%)	
	pART1	**ART1**
5	32.72	27.95
10	55.91	52.63
20	60.34	57.12
30	68.49	66.07
40	72.58	69.40
50	81.40	78.31

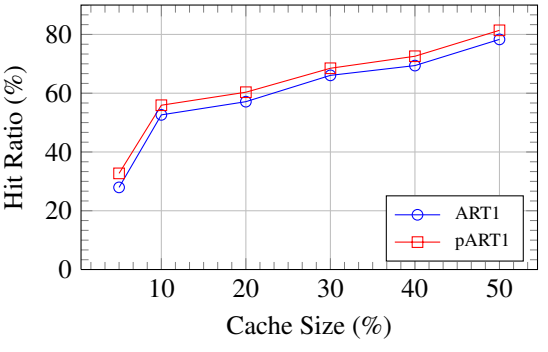


Fig. 7 Comparison of ART1 and pART1 (Trent Library data)

4.3 Effect of the Vigilance Parameter

One important parameter in ART1 is the vigilance parameter ρ. It decides the degree of similarity we want, so that a request pattern falls into a cluster. To investigate the effect of ρ to the performance of pART1, we selected its values as $0.4, 0.5, 0.6$ and 0.7 respectively and test using a cache size of 10% of the server size. Table 2 shows the collected results for three different scenarios of page requests discussed in Section 4.1. The column labeled "large chunk" shows that when ρ increases, cache hit also increases but very slowly when large chunk of pages are popular. It happens because there is very little similarities of request pattern in this type of input. But when a small group of pages are more popular (column "small chunk"), the hit rate increases more rapidly when the value of ρ increases. In the case of all pages have similar request probabilities, there is not much difference in hit rate when ρ is changed. The results are given in the column labeled "no pages".

In our experiment with the digital library data set, $\rho = 0.5$ because there was not a particular request patterns noticed. So we choose ρ in such a way which can deal with any pattern.

Table 2 Effect of vigilance parameter ρ

ρ	Cache Hit %		
	large chunk	small chunk	no pages
0.4	12.88	67.05	10.53
0.5	13.70	70.91	10.55
0.6	14.23	72.31	10.47
0.7	14.79	77.26	10.38

5 Conclusions and Future Work

Methods have been developed to realize the prefetching strategy based on various ideas for making prefetch decision, including the traditional ART1 neural network approach that can be used for clustering the web pages, such as the work [10]. However, since the input vectors were not dynamically generated in [10], their model only performed well for small systems. When considering big systems with over 10,000 different URLs, it will be very hard to use prefetching by that approach. The pART1 algorithm for prefetching we developed and described in this paper is an improvement of the ART1 algorithm to overcome this drawback. Our improvements have the advantage of constant high performance from small to large systems and different access patterns.

The simulation models developed in this paper mainly focus on the variation of cache size. Performance evaluation was based on cache hit rates. Some factors such as page size, content and network traffic were not considered. In further research, refined simulation models can be developed to include more factors.

Ackmowledgment. This project was partly supported by a grant from the NSRC (Natural Sciences Research Committee) of Trent University, Canada.

References

1. Aho, A.V., Denning, P.J., Ulmman, J.D.: Principles of optimal page replacement. Journal of the ACM 18(1), 80–93 (1971)
2. Carpenter, G.A., Grossberg, S.: Adaptive resonance theory. In: Encyclopedia of Machine Learning, pp. 22–35. Springer (2010)
3. Fausett, L.V.: Fundamentals of Neural Networks: Architectures, Algorithms and Applications. Prentice-Hall (1994)

4. Grossberg, S.: Adaptive pattern classification and universal recoding, I: Parallel development and coding of neural feature detectors. Biological Cybernetics 23, 121–134 (1976)
5. Hollmann, J., Ardö, A., Stenström, P.: Empirical observation regarding predictibility in user-access behavior in a distributed digital library system. In: Proceedings of the 16th International Parallel and Distributed Processing Symposium, pp. 221–228. IEEE Computer Society (2002)
6. Hollmann, J., Ardö, A., Stenström, P.: An Evaluation of Document Prefetching in a Distributed Digital Library. In: Koch, T., Sølvberg, I.T. (eds.) ECDL 2003. LNCS, vol. 2769, pp. 276–287. Springer, Heidelberg (2003)
7. Hollmann, J., Ardö, A., Stenström, P.: Effectiveness of caching in a distributed digital library system. Journal of System Architecture 53, 403–416 (2007)
8. Jones, S., Cunningham, S.J., Mcnab, R., Boddie, S.: A transaction log analysis of digital library. International Journal on Digital Libraries 3(2), 152–169 (2000)
9. Kazi, T.H., Feng, W., Hu, G.: Web object prefetching: Approaches and a new algorithm. In: Proceedings of the 11th International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, pp. 115–120. IEEE Computer Society, London (2010)
10. Rangarajan, S.K., Phoha, V.V., Balagani, K., Selmic, R., Iyengar, S.S.: Adaptive neural network clustering of Web users. IEEE Computer 37(4), 34–40 (2004)
11. Tanenbaum, A.S.: Modern Operating Systems, 2nd edn. Prentice-Hall (2001)

A User-Adaptive Augmented Reality System in Mobile Computing Environment

Sejin Oh and Yung-Cheol Byun*

Abstract. In this paper, we present a user-adaptive augmented reality (AR) system that augments physical objects with personalized content according to user's context as well as preferences. Since a user prefers different content according to the context, it reasons the user's recent content preferences through artificial neural networks trained with the feedback history describing which content the user liked or disliked with respect to his/her context. The system recommends a set of content relevant to the user's context and preferences. Then, it enables the user to select a preferred content among the recommended set and superimposes the selected content over physical objects. We implemented a prototype illustrating how our system could be used in daily life and evaluate its performance. From experimental results, we could confirm that our system effectively assisted users through personalized content augmentation in mobile computing environment.

1 Introduction

With recent advances in mobile devices, there has been a corresponding increase in the number of mobile services available to enjoy and experience. In particular, these services support users in achieving tasks through information assistance, e.g., navigation [3, 6, 7, 10]. Moreover, researchers have developed augmented reality (AR) systems that allow users to interact with computer-generated content, without getting distracted from the real environments around them [1]. Most AR systems have focused on seamless information augmentation over physical objects.

Sejin Oh
Convergence R&D Lab., LG Electronics, Seoul, S. Korea
e-mail: sjin.oh@lge.com

Yung-Cheol Byun
Dept. of Computer Engineering, Jeju National University, Jeju, S. Korea
e-mail: ycb@jejunu.ac.kr

* Corresponding author.

R. Lee (Ed.): Software and Network Engineering, SCI 413, pp. 41–53.
springerlink.com

However, these systems sometimes disturb users by overlaying information irrelevant to their needs or preferences. To assist users through the augmentation effectively, current AR systems are required to understand what and how users want to be supported in their context.

Some researchers have studied on adaptive information augmentation with respect to the user's context or preference. Based on the UMAR framework, information retrieval systems fetched information relevant to the user's context and overlaid it differently according to the context [4]. The *MARS* introduced a new approach for content arrangement according to a user's location and orientation in AR systems [5]. Pervasive services were visualized by considering how to bind appropriate virtual 3D models to physical objects by reflecting user's preferences [8]. These works enabled users to experience superimposed content in a personalized manner; however, they did not exploit content augmentation according to both user's context and preferences. Although the preferences occasionally change in a given context, these works did not take into account the changes and adapt content to the user's recent preferences in the context.

In this paper, we present a user-adaptive AR system which gives context-aware content augmentation adaptable to the user's most recent preferences. Since each user presents different preferences according to the user's environment, the system predicts a user's context by interpreting sensory information from mobile sensors. It reasons content preferences through artificial neural networks (ANN) [9] trained with feedback history describing which content the user actually accessed or not according to the context. Our system selects relevant content with higher similarity score to the preferences, and overlays the content over physical objects. Since preferences change over time, it keeps on tracking the user's clicks as explicit content-related feedback, and retrains the ANN accordingly.

Our user-adaptive AR system features following advantages. When a user looks at a physical object through the camera embedded to a mobile device, it automatically reasons the user's most up-to-date content preferences in the user's context. Our system enhances the user's satisfaction by presenting the right content to the content preferences and enables the user to immediately access desirable content, avoiding useless browsing to select the content. Through experiments with a prototype on an ultra mobile PC (UMPC), we prove that our system can effectively support users with personalized content augmentation without overburdening them.

In the following sections, we detail the proposed user-adaptive AR system, and describe an interaction with a prototype. We verify the effectiveness of our AR system through experiments and conclude with remarks about our work.

2 A User-Adaptive Augmented Reality System

We describe an innovative AR system that offers personalized content according to users by understanding what they want to be supported by in their context. As requirements, this system must acquire and interpret information about a user's context, evaluate his/her content preferences depending on the context, and adapt

content accordingly. To meet these requirements, as shown in Fig. 1, the proposed AR system consists of four key components: Context Reasoner, User Preference Manager, Content Adaptor, and Content Renderer. The Context Reasoner acquires situational information from mobile sensors and infers a user's context. The User Preference Manager reasons content preferences in the current context by utilizing the feedback history. The Content Adaptor selects the set of content relevant to context and preferences. The Content Renderer determines how to present the content and delivers it through augmentation over physical objects.

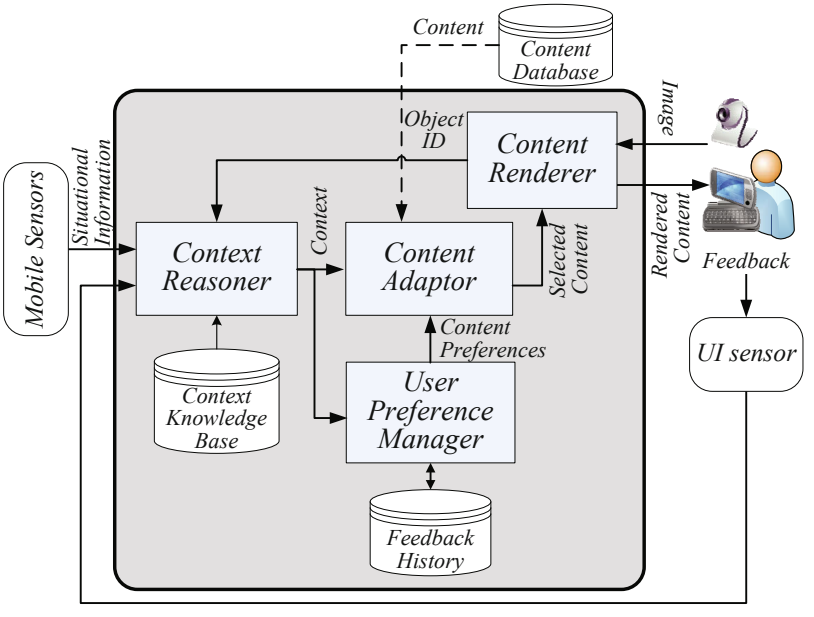


Fig. 1 System architecture

2.1 Context Reasoning on a Mobile Device

A system offering context-sensitive content to a user through a mobile device must be aware of the user's context and adapt content to changes in context. First, it must acquire situational information from sensors. Due to the diversity of possible sensors in the environments and mobile devices, it must interpret heterogeneous information to analyze complex contexts, e.g. a behavior in a certain location. Moreover, interpreting high-level context from low-level context is essential to build mobile applications with context-awareness.

Here, we specify a sensor as an entity that detects changes related to a user's context and converts this information into a description of the context. We assume that each sensor contains a context wrapper transforming a sensed signal into a XML

description. Practically, the sensor can be physical or virtual. A physical sensor, e.g., a camera and a location tracking sensor, can detect a user's basic context (e.g. visible objects, location). A virtual sensor, e.g., a content viewer or scheduler, can extract the user's semantic context (e.g. viewed content, upcoming event). For instance, let's imagine that the user Jack is near a desk and browses a list of images provided by the content viewer of a mobile device. In this case, an indoor location tracking sensor detects his current location and generates a description indicating that he is located at the desk. Similarly, the content viewer generates a description of the situation indicating that he viewed some specific content.

The proposed system perceives a user's context through context acquisition and reasoning. The Context Reasoner acquires a set of situational information from several mobile sensors. It aggregates the collected information to understand the user's current context; it groups similar elements and extracts one group consisting of the largest number of elements in a given dimension. Based on the merged information, the Context Reasoner infers abstract, high-level context (e.g., what does the user intend to do?) by exploiting user-defined context reasoning; application developers must provide horn-logic rules dedicated to their particular applications. The rules are stored in a context knowledge base (KB). Finally, the Context Reasoner generates the description of the user's context.

2.2 User Preference Management

To offer the right content to users according to their context, our system predicts which content they prefer and how to render them according to the context. The description of context from the Context Reasoner, the User Preference Manger converts it to user's feedback consisting of the user's context, the content presented to the user, and feedback value for the content. Since the same user differently appreciates the same content according to the context, the User Preference Manager evaluates the relationship between feedbacks and contextual factors using an artificial neural network (ANN). As preferences in a given context may change, it retrains the ANN periodically.

The User Preference Manager extracts content-related feedback from the user's clicks regarding content. When some content is presented to a user on a mobile device, three click-based selections are available: display, ignore, and remove. In the absence of selection within a given time, the content matching the preferences best is automatically displayed. The user can ignore the content and request another, absent from the recommendations. Content clicked by the user are considered most relevant. Inversely, content removed by the user are considered least relevant. Therefore, according to a user's clicks regarding content C_i in a context CO_x, the effect $E_{C_iCO_x}$ is adjusted by rules in Eq. 1. The User Preference Manager estimates new feedback value associated with the clicked content by using Eq. 2, and updates the feedback history.

$$E_{C_iCO_x} = \begin{cases} +2\alpha & \text{if a user selects content } C_i; \\ +\alpha & \text{if } C_i \text{ is selected automatically;} \\ -\alpha & \text{if a user ignores } C_i; \\ -2\alpha & \text{if a user deletes } C_i. \end{cases} \quad (0 < \alpha) \tag{1}$$

where C_i is offered content and α is the scale factor for feedback values.

$$f_{C_iCO_x}(t) = (1-\sigma) \times f_{C_iCO_x}(t-1) + \sigma \times E_{C_iCO_x} \quad (0 < \sigma \leq 1) \tag{2}$$

where $f_{C_iCO_x}(t)$ is the updated feedback value of content C_i in the context CO_x and $f_{C_iCO_x}(t-1)$ is its previous feedback value. If there is no existing data associated with the extracted feedback for C_i in context CO_x, it sets $f_{C_iCO_x}(t-1)$ to 0. The σ is the learning rate indicating how quickly the new feedback supersedes the old. $E_{C_iCO_x}$ is computed from explicit profiling in Eq. 1.

The User Preference Manager predicts context-sensitive content preferences by analyzing relationships between the user's liked/disliked content and context. Since contextual factors affect to content preferences in a non-linear manger, we propose preference reasoning employing artificial neural networks (ANN), similar to a human brain consisting of multilayered perceptrons (MLPs) [9]. The perceptron is composed of an input layer, one or more hidden layers, and an output layer. As shown in Fig. 2, to support the reasoning of the user's up-to-date preferences, we let the User Preference Manager relearns the inference network by exploiting the feedback history, i.e. links between a user's context and the favorite content in the context, periodically. Here, the User Preference Manager places the contextual factor is the input layer and content preferences including the description of content and its feedback value in the output layer, respectively, During the given iterations, ANN is updated on the basis of back-propagation algorithm. Then, the User Preference Manager derives appropriate content preferences by the trained ANN with respect to the user's perceived context.

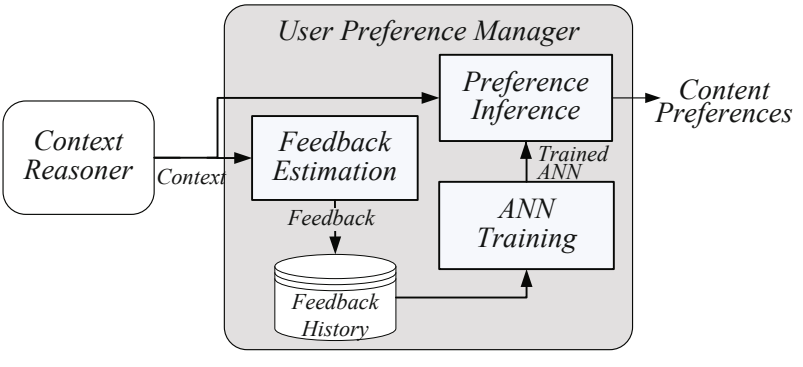


Fig. 2 Training of the ANN with feedback history.

2.3 Personalized Content Augmentation

Our system recommends relevant multimedia content to reasoned content preferences. It filters the available content, displays it as a list on a mobile device, and enables the user to select one of them. As shown in Fig. 3, the Content Adaptor retrieves the content relevant to the user's context from the content database. To select the content suitable for the user, it determines the degree of relevance between content preferences and the retrieved content. Then, it generates the list of content relevant to the user's preference in the context. Then, the Content Renderer determines how to represent the selected content with respect to the existence of associated objects and visualizes it in an appropriate presentation.

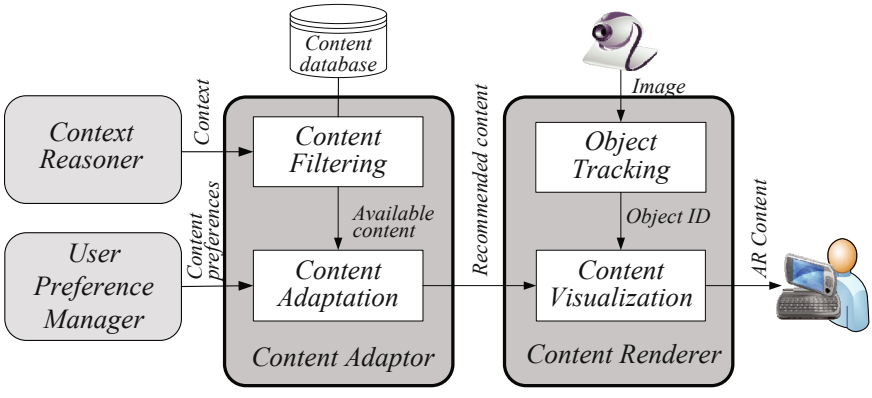


Fig. 3 Content customization and augmentation adaptable to a user's context and content preferences.

Content preferences are represented as a vector consisting of the set of 2-tuple <feature, weight>. Each term is described as the combination of the content type and the supported function; the feedback is the numerical value indicating how much a user likes or dislikes a given content. When the preferences contain different terms, they are represented as $P = \{(t_1, f_1), (t_2, f_2), .., (t_m, f_m)\}$. Each content available for the current context is also described as a vector $C = \{(t_1, u_1), (t_2, u_2), .., (t_m, u_m)\}$ where t_i is the same in the preference vector and u_i is the weight assigned to term t_i. Since terms are not all equally important to filter content, we assign them relative importance according to the field. We define the field set $S = \{type, function\}$, since each term is composed of these two fields, i.e. content type and the supported function. Based on $W = \{W_s | s \in S\}$ where W_s is the importance factor assigned to terms in field s, the weight u_i is assigned by using Eq. 3.

$$u_i = \begin{cases} \sum_{k=1}^{N} W_s & \text{if } t_i \text{ includes } N \text{ number of fields; } (N \leq 1) \\ 0 & \text{if } t_i \text{ does not include any field.} \end{cases} \tag{3}$$

The Content Adaptor evaluates the degree of relevance between the preferences and available content for the context: it determines the similarity between the vector for content preferences and that for the retrieved content. Such similarity is measured using cosine of the angle between the preference vector P and the content vector C as shown in Eq. 4. In addition, it employs a threshold to determine content with absolute correlates greater than a given value $\theta(0 < \theta \leq 1)$. Finally, it sorts the list by decreasing the similarity.

$$similarity(P,C) = \frac{P \times C}{||P|| \times ||C||} = \frac{\sum_{i=1}^{m} w_i u_i}{\sqrt{\sum_{i=1}^{m} w_i^2 \sum_{i=1}^{m} u_i^2}} \tag{4}$$

The Content Renderer adaptively visualizes the content on a mobile device. When a user looks at the object through a camera in a mobile device, it automatically analyzes images acquired from the camera to determine the existence of the objects [13]. Then, the Content Renderer displays the set of content recommended by the Contend Adaptor and enables a user to select a preferred content among the recommended set. It determines the proper placement of the content according to the existence of associated physical objects in the camera's view. If the related object exists, it estimates a camera pose, i.e. the relative position of an associated object in the camera's viewpoint. Then, the Content Renderer overlays the selected content over the object. To support continuous content augmentation, it tracks the object with an image-to-image tracking method [2]. If there is no related physical object, it displays the content on the screen of the mobile device instead.

3 Implementation

To verify the usefulness of our system, we developed a prototype illustrating support for users through personalized content augmentation in our daily life. We implemented it on an ultra mobile PC (UMPC) with a 1.33 GHz CPU and 1GB of RAM. The implemented prototype superimposes multimodal content adapted to the user's context and preferences in real-time and exploited the *OpenScenegraph* to render the content on the UMPC [11]. Here, we limited the content to images, texts, sounds, movies, and 3D models. The prototype enabled a user to experience personalized content augmentation over physical objects. Fig. 4 shows an example of the content augmented over a book by our prototype. It kept on offering personalized content augmentation whenever the user's context changed.

The prototype is aware of a user's context thanks to context aggregation and inference on the UMPC. Here, a camera embedded in the UMPC recognizes a physical object targeted by the user. To recognize physical objects, the prototype acquired a video sequence from the camera embedded in the UMPC: 30 frames of 320 by 240 pixels per second and employed an approach for natural feature-based object recognition and tracking. A touch sensor detects the user's feedback regarding the content. Indoor location tracking sensors track a user's current location. The implemented system continuously collected information from these sensors, integrated

(a) **(b)**

Fig. 4 The prototype: (a) A user's experience with the prototype and (b) the content augmented over a book.

it, and evaluated the user's intention per 30ms (to support 30 frames per a second for content rendering) with predefined rules from the contextual knowledge base as shown in Fig. 5.

```
(?user User#2)^(?user locatedIn ?table)^(?user watch ?book#3)^
(time ?morning)
→(?user intendTo ?relax)
(?user User#2)^(?user locatedIn ?table)^(?user watch ?book#2)^
(time ?afternoon)
→(?user intendTo ?study)
(?user User#2)^(?user locatedIn ?sofa)^(?user watch ?book#3)^
(time ?afternoon)
→(?user intendTo ?relax)
...
```

Fig. 5 Example of rules to infer the user's intention using location, an visible object, and time).

Before reasoning the user's content preference in a given context, the User Preference Manager constructs an ANN from feedback history describing which content the user likes or dislikes in a certain context. The feedback history consists of contextual factors (i.e., time, object, user's location, and intention) and content preferences described by a content type, a supported function, and a feedback value. As shown in Fig. 6, the User Preference Manager maps the contextual factors to the input layer, and associates content preferences to the output layer. It constructs nodes in a hidden layer as three times of the number of nodes in the input layer. The User Preference Manager sets the activation function $f(x) = tanh(x)$ in each node.

Weights associated with nodes are set randomly when the ANN is first trained. During training iterations, the weights are adjusted through back-propagation. Finally, the User Preference Manager predicts the user's content preferences for the current context based on the trained ANN. To improve the accuracy of reasoned results, it filters out irrelevant results with a confidence rate below 99%. To reason the user's recent preferences, the ANN is periodically retrained with the updated feedback history.

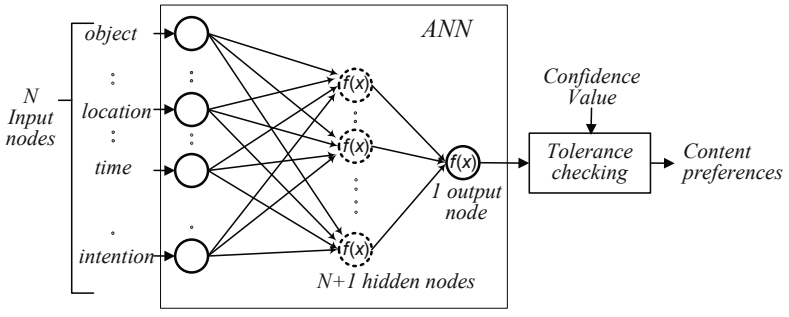
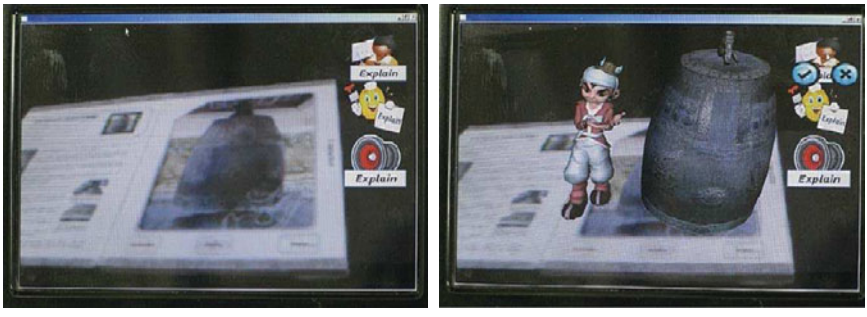


Fig. 6 Reasoning of content preferences based on the trained ANN.

The set of content are filtered and rendered on the UMPC according to the context-sensitive preferences. The Content Adaptor extracts the list of available content from content database, and then calculated the similarity between the retrieved content and preferences. The list of content with higher similarity is displayed (Fig. 7a). The user can select or remove the certain content in the list. When a user selects one, it is superimposed over its associated physical objects or simply displayed on the screen (Fig. 7b). If the user removes one from the list, the prototype renews the list with alternative recommendations. The information from touch interaction is transmitted to the Context Reasoner, and the User Preference Manager updates the feedback history. Based on the interaction loop, the prototype learns one's most recent preferences and offers content reflecting them.

4 Experimental Results

To show the effectiveness of the proposed user-adaptive AR system, we evaluated the factors with our prototype: how correctly it reasons content preferences, and how properly it adapts content to the context and preferences. 15 participants experienced our prototype: 9 male and 6 female university students in their twenties knowledgeable about other AR systems. Here, we built data sets consisting of the preferences established by 15 users. During experiments, the implemented prototype was used to overlay the content in eight situations with different location, time, and objects. Then, we recorded from 180 to 240 user-system interactions for one



(a) (b)

Fig. 7 Content augmentation: (a) the list of recommended content for a book and (b) rendering the selected content over the book.

week period per each data set. Each interaction contained the problem-solving context, the content offered by our system and user feedback value to the content.

4.1 An Evaluation of Preference Reasoning

We evaluated the accuracy of the preference prediction by comparing our ANN-based reasoning with a decision tree-based approach. Since ID3 decision tree-based classification is one of the most widely used methods for inductive inference [12], we built an ID3-based decision tree from the recorded feedback history. We verified the appropriateness of reasoned preferences with these two algorithms by estimating correctness for each circumstance as follows.

$$Correctness = \frac{N_{relevant}}{N_{reasoned}} \tag{5}$$

where $N_{reasoned}$ is the number of reasoned preferences, and $N_{relevant}$ is the number of relevant ones.

Fig. 8 shows the performance of these two algorithms for individual data sets; the results for ID3 are ranked by increasing average correctness, and are presented along with the results for ANN-based reasoning. The last set mirrors their overall performance. For example, the first pair of bars indicates that for user 2, our approach correctly predicted more than 80% of his preferences, on the other hand, ID3-based decision reasoned about 65% of the preferences. Overall, our ANN-based reasoning seems more accurate than ID3-based reasoning (about 0.85 about 0.77). In particular, our approach proved significantly superior when they did not remarkably show different content preferences in different circumstances (User 2, 5, 10, and 15). Since the ID3-based approach tests a single attribute at a time for decisions, it thus cannot construct a good decision tree. Since the ANN-based approach simultaneously considers multiple contextual factors for classification, it

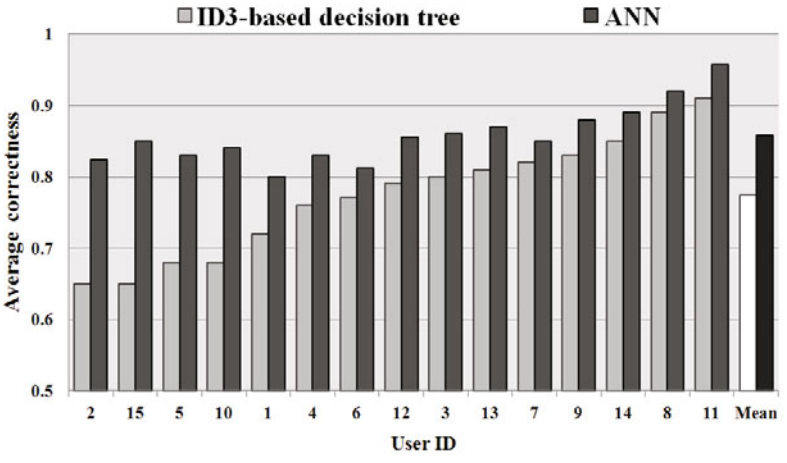


Fig. 8 Comparative performance of an ID3-based decision tree approach and our ANN-based reasoning.

outperforms ID3-based reasoning. Hence, our approach more effectively predicts content preferences for users with context-sensitive preferences or consistent preferences.

4.2 An Evaluation of Content Adaptation

We observed the system's ability to correctly offer content in each context. In this experiment, the participants performed in the 8 situations in two conditions: 1) with respect to the context, 2) regardless of the context. We compared the performance between these two conditions. To estimate the correctness of the offered content, the participants gave feedback, i.e., direct or automatic selection, neglect, and removal, through clicks on the UMPC. Participant removed or ignored unsuitable content which were marked as negative. Based on this rating method, we evaluated the performance of the system by using Eq. 6. The means for these two conditions were also analyzed by t-test for paired samples.

$$Quality = \frac{N - N_{neg}}{N} \tag{6}$$

where N is the number of items on the list of offered content, and N_{neg} is the number of items on the list of offered content with a negative score.

In Fig. 9, the differences between the average qualities in both conditions for the eight situations are presented in ascending order; the last set mirrors their overall qualities. Overall, predicting preferences from the context worked better than without this cue (about 0.85 against 0.75). Our system was significantly more effective to assist participants with context-related preferences. For example, one

participant frequently requested explanations of the traditional bell with a 3D model but preferred textual explanations with others; our system understood that and proposed text in the condition "with context-based adaptation", however, the condition regardless of the context always proposed the 3D model-based explanations. In the other condition, when participants showed consistent preferences in different circumstances (User 2, 5, 10, and 15), the performances were similar in both conditions. Therefore, our system more effectively adapted content by using one's preferences and context, with a gain depending on his/her specificities

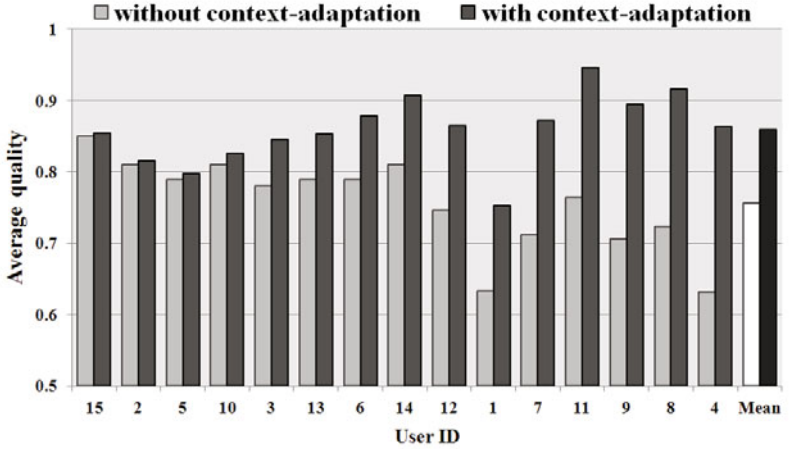


Fig. 9 Average quality of content adaptation without and with considering the context.

5 Conclusion and Future Work

We presented a user-adaptive AR system that offers context-sensitive content augmentation adapted to user preferences in real-time. The system observes a user's context and reasons which content the user wants to be supported by in the context. It enables the user to experience desired content in customized ways. Although we experiment with a small group, our user-adaptive AR system correctly estimates content preferences for the user's context, and effectively customizes content to their context and preferences. Finally, we confirm that content personalization based on context awareness and user preference reasoning is effective to offer personal assistance in AR systems.

Because reasoning fails in new situations for lack of appropriate feedback history, we need preference reasoning mechanisms for new circumstances. Since some users refuse to give feedback explicitly, we plan an implicit approach by analyzing the user behavior patterns. Moreover, we should extend current approaches for context awareness and preference reasoning to long-term uses in mobile computing environment.

Acknowledgments. This research was financially supported by the Ministry of Education, Science Technology (MEST) and Korea Industrial Technology Foundation (KOTEF) through the Human Resource Training Project for Regional Innovation.

References

1. Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B.: Recent Advances in Augmented Reality. IEEE Computer Graphics and Applications 21(6), 34–47 (2001)
2. Baker, S., Matthews, I.: Lucas-Kanade 20 years on: A Unifying Framework. International Journal of Computer Vision 56(3), 221–255 (2004)
3. Cheverst, K., Davies, N., Mitchell, K., Smith, P.: Providing Tailored (context-aware) Information to City Visitors. In: Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, pp. 73–85 (2000)
4. Henrysson, A., Ollila, M.: UMAR: Ubiquitous Mobile Augmented Reality. In: Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia (MUM 2004), pp. 41–45 (2004)
5. Höllerer, T., Feiner, S., Hallaway, D., Bell, B., Lanzagorta, M., Brown, D., Julier, S., Baillot, Y., Rosenblum, L.: User Interface Management Techniques for Collaborative Mobile Augmented Reality. Computer Graphics-UK 25(5), 799–810 (2001)
6. Kim, C., Lee, J., Cho, Y., Kim, D.: VISCORS: A Visual-Content Recommender For The Mobile Web. IEEE Intelligent System 19(6), 32–39 (2004)
7. Lee, W., Wang, J.: A User-centered Remote Control System For Personalized Multimedia Channel Selection. IEEE Transactions on Consumer Electronics 50(4), 1009–1015 (2004)
8. Lee, J., Seo, D., Rhee, G.: Visualization and Interaction of Pervasive Services using Contextaware Augmented Reality. Expert Systems and Applications 35(4), 1872–1882 (2008)
9. Lippmann, R.P.: An Introduction to Computing with Neural Nets. IEEE ASSP Magazine 4(2), 4–22 (1987)
10. Liu, N., Kung, H.: JoMP: A Mobile Music Player Agent For Joggers Based on User Interest and Pace. IEEE Transactions on Consumer Electronics 50(4), 1009–1015 (2009)
11. OpenScenegraph, http://www.openscenegraph.org/projects/osg
12. Quinlan, J.R.: Induction of Decision Trees. Machine Learning 1, 81–106 (1986)
13. Rosten, E., Porter, R., Drummond, T.: Faster and Better: A Machine Learning Approach to Corner Detection. IEEE Transactions on Pattern Analaysis and Machine Intelligence 32(1), 5–119 (2010)

Software Development Education Based on UEC Software Repository

Takaaki Goto, Takahiro Homma, Kensei Tsuchida, and Tetsuro Nishino

Abstract. Most of the software developed in universities is primarily used by the developers themselves. Normally, most of this software is managed and stored on servers in the research laboratories, but since the software is generally lacking in documentation and is not developed with third-party use in mind, it tends to be used only by the original developers. It is seldom reused after the developers have graduated, and is often not in a fit state for use by third parties. Today's information systems graduates have not been provided with sufficient education with regard to the knowledge, techniques and experience needed for the usual software development process of actual software development businesses from project planning through to execution and management (requirements analysis, development, implementation, testing, etc.) and lack the basic skills for handling actual business situations. In this paper, we report on our approach to software management using the UEC software repository to consolidate software assets, and on practical software development education based on this repository.

Takaaki Goto · Takahiro Homma
Center for Industrial and Governmental Relations,
The University of Electro-Communications,
1-5-1 Chofugaoka, Chofu-shi, Tokyo, Japan
e-mail: {goto,homma}@kikou.uec.ac.jp

Kensei Tsuchida
Faculty of Information Science and Arts, Toyo University,
2100 Kujirai, Kawagoe-shi, Saitama, Japan
e-mail: kensei@toyo.jp

Tetsuro Nishino
Graduate School of Informatics and Engineering,
The University of Electro-Communications,
1-5-1 Chofugaoka, Chofu-shi, Tokyo, Japan
e-mail: nishino@ice.uec.ac.jp

R. Lee (Ed.): Software and Network Engineering, SCI 413, pp. 55–65.
springerlink.com

1 Introduction

Institutional repositories are currently under construction at over a hundred universities in Japan [1]. An institutional repository is a place where an organization such as a university stores and publishes the results of its research, such as research papers, research reports and research bulletins. However, institutional repositories are not currently treated to store software produced by universities.

Repositories that can handle software include folio direct and others [2]. SourceForge [3] is repository used for open-source software that not only provides facilities for publishing computer programs but also supports software development by offering additional features for managing program versions, operating bulletin boards and so on. And also github supports collaborative software development [4].

At the University of Electro-Communications, a great deal of software is written every year through the course of student research and educational activities. Normally, most of this software is managed and stored on servers in the research laboratories, but since the software is generally lacking in documentation and is not developed with third-party use in mind, it tends to be used only by the original developers. Once the developers have graduated, there are very few opportunities for this software to be reused. As one approach to addressing these problems, we are working on the construction of a UEC software repository. The UEC software repository is a database for the centralized in-house management of software developed at the university. Our aim is to make it possible for ordinary users to download and use software stored in this repository by accessing it through site searches.

Meanwhile, advances in telecommunications technology are turning the software industry into one of the key industries of the 21st century. However, in recent years Japan's IT industry has experienced a very sharp decline in technical ability and creativity. According to economic organizations such as the Japan Business Federation and other related companies, a major cause of this decline is the mismatch between the educational needs of corporations and the information systems education provided by graduate schools. [5]. For example, today's information systems graduates have not been provided with sufficient education with regard to the knowledge, techniques and experience needed for the usual software development process of actual software development businesses from project planning through to execution and management (requirements analysis, development, implementation, testing, etc.) and lack the basic skills for handling actual business situations. The key to eliminating the human resources mismatch that occurs between industry and academia is to cultivate IT professionals that can meet the demands of industry. To resolve this serious situation, we need to reinforce the parts of the curriculum that are currently lacking.

In this paper, we report on the construction of the UEC software repository, and on our efforts to take a more practical approach to software development education with the UEC software repository at its core.

2 The UEC Software Repository

2.1 Overall Concept

Figure 1 illustrates the concept of the UEC software repository.

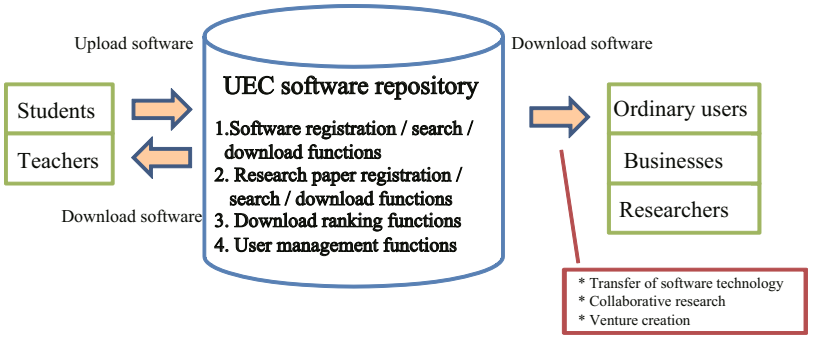


Fig. 1 Conceptual illustration of the UEC software repository

Software developed by students and teachers is uploaded to and registered in the repository. When registering software in the repository, the person registering the software can choose to either retain the software copyright or to transfer it over to the university. Software published in the UEC software repository is not only available for download by students and teachers, but is also made publicly available for downloading by ordinary users, businesses, researchers and the like.

2.2 System Overview

The repository system has the following features:

1. Software registration, search and download functions
2. Research paper registration, search and download functions
3. Download ranking functions
4. User management functions
5. Development support functions

The user registration and software registration procedures are discussed below.

2.2.1 User Registration

This repository supports three types of user with different privileges:

1. Software project leader
2. Software users
3. System administrators

Table 1 lists the features available to each type of user.

Table 1 Features available to each user type

Software project leader	Registering, searching and downloading software; searching and downloading research papers
Software users	Searching and downloading software; searching and downloading research papers
System administrators	Managing software, managing research papers, managing users

Software project leaders are permitted to submit software to the repository. To qualify as a software project leader, a user must have studied for and completed a graduate course in the practical software development course, which is described below. Software project leaders are also able to search and download research papers.

Software users are permitted to search and download the software and research papers published in the repository. Software users need no particular qualifications, and only need to complete the software user registration process in the repository.

System administrators are permitted to check and approve software that has been submitted to the repository, and to perform user administration. They are also permitted to register and manage research papers such as technical reports.

As an example, Fig. 2 illustrates the registration procedure for software users. To register as a software user, the user first accesses the repository and clicks the

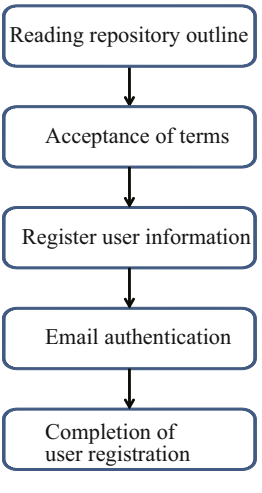


Fig. 2 Software user registration process

user registration link. The user is then shown an outline document that describes the purpose of the repository, and a simple copyright statement. When the user clicks to accept this information and move to the next page, a UEC software repository user agreement is displayed. At the bottom of this user license, the user is asked to mark a check box to confirm his or her acceptance of items of particular importance in the user agreement, such as the member eligibility conditions and prohibited actions. Next, the user provides a login ID and password together with details such as the user's name and email address. When the user has finished registering his or her user information in the repository, the system sends out a confirmation email containing a link that the user must click to complete the registration process. A similar procedure is used for the registration of software project leaders.

2.2.2 Software Registration

Registration of software in the repository can only be performed by users that are eligible to act as software project leaders. When registering software in the repository, software project leaders are able to choose either of the following copyright conditions for software they have developed:

1. Transfer copyright to the university
2. Keep copyright with project leader

It was originally envisaged that all copyrights on software registered in the repository would be transferred from the software project leader to the university at the time of registration, and that the university would then grant users the right to freely copy and adapt software for their own purposes. However, when the teaching staff were surveyed to find software candidates for inclusion in the repository, we found many cases where they were unwilling to transfer copyright to the university. We therefore adopted a system where software project leaders can choose either to transfer their copyright to the university or to retain their individual copyright without transferring it to the university.

Figure 3 illustrates the procedure for submitting software to the repository. First, the software project leader logs in to the repository. This brings up a software registration page where the project leader registers the details of the software. These details include the name of the software, its scope of use, its version number, a list of keywords, a summary, a description, and a list of co-authors. The project leader can then upload the software data and summary documentation, at which time the items relating to software registration in the user agreement are displayed once again and the project leader must confirm these items by marking a checkbox again before completing the upload process. When an upload is processed, the software specified in the registration procedure becomes provisionally registered in the repository.

After the software project leader has provisionally registered the software in the repository, the application details are checked by a repository system administrator. This includes checking the information submitted together with the software, and checking the software for viruses. All being well, the work of registering the software is continued. The next step is the copyright transfer procedure, although this

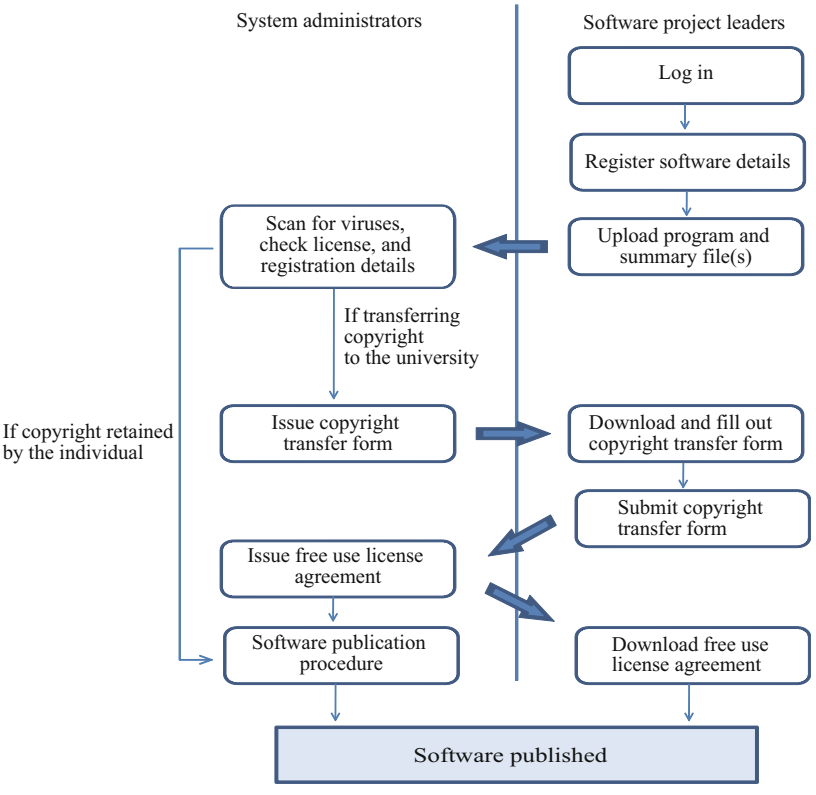


Fig. 3 Software submission procedure

is unnecessary if the software copyright is being retained by the project leader. The system administrator then performs the software publication process to publish the software. When the software copyright is transferred to the university, a copyright transfer form is displayed in the page showing the details of this software in the software project leader's user pages, from where it can be downloaded. The software project leader downloads the copyright transfer form from the repository, adds a signature and/or seal, and submits it to the university. Once the university has received the copyright transfer form from the software project leader, the system administrator issues a free use license agreement on the repository, and the software project leader becomes able to download it from his or her user pages. Finally, the system administrator performs the software publication process to publish the software.

Then the project leader is given a project page of its software to collaborate with coauthor. A project page has functions such as forums, bug tracking, and so on. Project leaders can treat their project page either public or private.

3 A Practical Approach to Software Development Education

We are working on a more practical form of software development education that provides students with a more rounded set of practical skills and self-motivated qualities in software development. In this effort, by developing teaching materials for self-motivated practical education centered on open-source software development and implementing an education program that incorporates these materials, we aim to provide practical education using the UEC software repository as an autonomous practical educational resource in order to cultivate highly creative developers with excellent research and development skills.

3.1 Overall Concept

In this section, we describe our efforts to take a more practical approach to the teaching of software development. Figure 4 shows the overall organization of these efforts.

Our efforts are centered on the UEC software repository and a course in practical software development. In the practical software development course, students receive lectures on software engineering, system infrastructure, and OSS development. On completing these lectures, they are granted permission to make submissions to the UEC software repository. After receiving permission to submit software to the UEC software repository, the students continue with their PhD or MSc

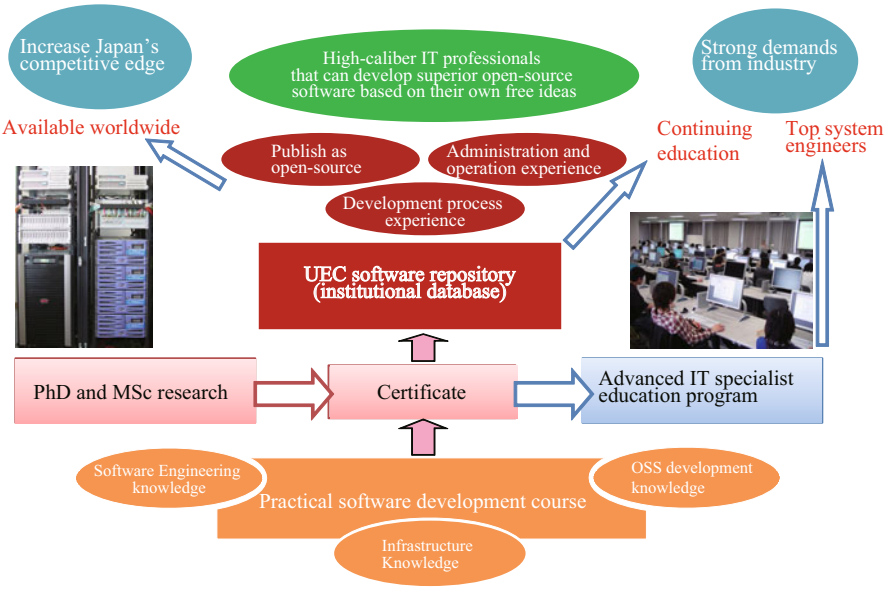


Fig. 4 Autonomous practical education of open-source software development skills

studies, and publish their research results in the repository as open-source software. By making practical use of the system development knowledge gained in the practical software development course in their research and development work and submitting the resulting software to the repository, students are able to experience the software development process for themselves.

By centering the education of graduate students around the UEC software repository and a practical software development course, we will instill our students with self-motivation and practical abilities.

3.2 Development of Teaching Materials for the Practical Software Development Course

In the development of teaching materials for the practical software development course, we plan to design courses on software engineering, infrastructure system architecture and OSS development. Some of the material have already designed, and we are continuously developing the lecture plans.

3.2.1 Fundamentals of Practical Software Development

The ”Fundamentals of Practical Software Development” module is concerned with the details of software engineering. This module is based on the results of the module “Introduction to practical software development I“ (Introduction to software engineering) which was conducted in the 2009-10 academic year, and in the 2010-11 academic year it was offered as an interdisciplinary specialist module “Fundamentals of Practical Software Development“ at the graduate school. For the introduction to practical software development, in addition to an overview of software engineering, a larger proportion of time was set aside for students to engage in proactive tasks such as discussing a chosen theme or developing Web applications, resulting in a course with greater practical content that more closely resembles the scene of actual software development. Since the course includes Java software development exercises, it is necessary for students to have previously gained the requisite knowledge of “Basic Java“ relating to basic details such as classes and inheritance equivalent to an introductory course in Java, and “Applied Java“ relating to the construction of Web applications using server-side Java.

3.2.2 Foundations of IT System Software

In the ”Foundations of IT System Software” module, students will perform hands-on learning of infrastructure systems such as operating systems, networks and databases (Linux, Tomcat, PostgreSQL etc.) in order to gain a thorough understanding of the workings of the UEC software repository. And also students can learn the working principle of infrastructure systems systematically, and have to construct a server in consideration of performance of its system.

3.2.3 Foundations of Open Source Software Development

In the ”Foundations of Open Source Software Development” module, students will engage in PBL (Problem-Based Learning) to develop OSS for inclusion in the UEC software repository. We obtained several requests about development theme from our campus and neighboring school. Tools such as a matrix calculator, ledger sheet printer, a supporting tool of color amblyopia, and so on, are developed by students. In development phases, students can develop systems with their users; therefore it is a precious experience for students. All the OSS produced in this module will be submitted to the repository.

Students will also be shown how to prepare diverse English-language documentation, and will experience OSS development in English with a team including foreign students.

3.3 Evaluations

We evaluated one of our lectures, “Fundamentals of Practical Software Development,“ by conducting a questionnaire survey at the end of it. The survey was a multiple-choice questionnaire and answered by 53 out of a total 60 students. Table 2 shows the results of the questionnaire.

One question was “Did you obtain knowledge, the ability to think, and skills from this lecture¿‘ Over half of the students answered that they had obtained them. The other question was “Do you think that you were satisfied with the lecture¿‘ About 42% and 30% answered “strongly agreed“ or “agreed,“ respectively. Because of these responses, we think our course meet the students’ requests. We can also see an improvement in the result of this year’s questionnaire compared to previous fiscal year‘s results.

Table 2 Results of the questionnaire

Questions asked	SD (%)	D (%)	N (%)	A (%)	SA (%)
Did you obtain knowledge, the ability to think, and skills from this lecture?	1.9	5.7	26.4	35.8	30.2
Do you think that you were satisfied with the lecture?	5.7	5.7	17	16	41.5

SD: Strongly Disagree, D: Disagree, N: Neutral, A: Agree, SA: Strongly Agree.

4 Related Works

Much research has been done on software engineering education such as [6, 7, 8, 9].

[10] reports an education of software engineering education using software repository. In this paper, they make good use of software repository in their lectures and research, especially for designing novel information visualization using its material such as “Description,“ “Applications,“ “How to Use,“ and so on. Our program is

weighted toward providing practical software development opportunities. It is important to develop software in consideration of evaluation by third parties. Students release their software developed in the lecture or research through the UEC software repository.

As for software development education using open source software, [11] is an example of it, describing a case in which students took part in an actual open source project, "NetBeans." We are also proceeding with research on the software development education for open source software. However, we have not adopted a plan to participate in open source project. We have developed some software to fulfill requests received from neighborhood schools. The ratio of Japanese student developers of open source project is low because of language difficulties; we hope that our lecture becomes an opportunity to get more involved in open source software development.

5 Conclusion

In this paper, we have presented an overview of the UEC software repository, and we have described our efforts to take a more practical approach to software development education centered on this repository.

Much of the software developed by students at UEC is of a very high level by international viewpoint, but since it mostly comprises highly specialized software and software with highly specialized usage methods, it is difficult for third parties to make use of this software. By using the UEC software repository to make this software publicly available so that it can be used by third parties, development techniques will be refined from a personal level to an international level, and students will learn by themselves about methods for producing high-quality user-friendly software. Also, by releasing to the world the high-quality software developed by students as a by-product of their education, it will be possible to provide software that is very useful to society in general.

In the future, we will promote the active use of UEC's software resources, and through our practical approach to the teaching of software development, we hope to continue making refinements to this framework to improve the quality of software developed by students.

References

1. National Institute of Informatics. NII Institutional Repositories Program, http://www.nii.ac.jp/irp/en/
2. Folio direct, http://www.foliodirect.net/
3. SourceForge, http://sourceforge.net/
4. Github, https://github.com/
5. Keidanren, http://www.keidanren.or.jp/japanese/policy/2005/039/index.html (in Japanese)

6. Almi, N., Rahman, N., Purusothaman, D., Sulaiman, S.: Proceedings of 2011 IEEE Symposium on Computers & Informatics (ISCI 2011), pp. 542–547 (2011), doi:10.1109/ISCI.2011.5958974
7. Barzilay, O., Hazzan, O., Yehudai, A.: IEEE Transactions on Education 52(3), 413 (2009), doi:10.1109/TE.2008.930094
8. Yadav, S.S., Xiahou, J.: Proceedings of 2010 International Conference on Educational and Network Technology (ICENT), pp. 34–36 (2010), doi:10.1109/ICENT.2010.5532120
9. Li, N., Liang, Q., Peng Zhang, Z.: Proceedings of 2010 Second International Workshop on Education Technology and Computer Science (ETCS), vol. 1, pp. 781–784 (2010), doi:10.1109/ETCS.2010.104
10. Borner, K., Zhou, Y.: Proceedings of Fifth International Conference on Information Visualisation 2001, pp. 257–262 (2001), doi:10.1109/IV.2001.942068
11. Jaccheri, L., Osterlie, T.: Proceedings of First International Workshop on Emerging Trends in FLOSS Research and Development (FLOSS 2007), p. 5 (2007), doi:10.1109/FLOSS.2007.12

A Winner Determination Method on GlobalAd Service: Model and Formulation

Satoshi Takahashi, Yuji Hashiura, Roger Y. Lee, and Tokuro Matsuo

Abstract. The Global Ad is a system to be used when users apply to publish their advertisement on newspapers. The advertisement is registered and uploaded on database that can be viewed by advertising agency. In actual use by multiple users, the system needs to allocate advertising spaces on the newspaper and to calculate the cost to publish the advertisements. Also, the system need to solve a problem like a packing problem with publishing costs/bid value. To solve the problems, in this paper, we propose a winner determination method using 2-dimensional knapsack problem.

1 Introduction

An advertisement fee is one of primal incomes of some paper medias such as newspapers and magazines. The paper media provides some spaces, which is used for advertisements, in their newspapers or magazines, and sells the space to some advertisers, who want to put own company's advertisement on the newspaper. The advertiser choices an appropriate advertisement space by comparing efficiency and advertisement fee. In general, advertisement fees are monotone increasing by size

Satoshi Takahashi
Graduate School of Systems and Information Engineering,
University of Tsukuba, Tsukuba, Japan
e-mail: takahashi2007@e-activity.org

Yuji Hashiura · Tokuro Matsuo
Graduate School of Science and Engineering, Yamagata University,
Yonezawa, Japan
e-mail: hashiura2009@e-activity.org, matsuo@yz.yamagata-u.ac.jp

Roger Y. Lee
Software Engineering and Information Technology Institute,
Central Michigan University, Michigan, USA
e-mail: lee@cps.cmich.edu

R. Lee (Ed.): Software and Network Engineering, SCI 413, pp. 67–76.
springerlink.com

of the advertisement space. On the other hand, since electronic medias, such as Yahoo!, MSN and some electronic newspaper companies, is improved rapidly[6], the readers of paper medias is decreasing and also the value of advertisements put on the paper media is decreasing. Especially, many advertisers want to put own advertisement on the Internet searching site such as Google[2] and Yahoo![1]. There are two reasons of this; one is lower advertisement fee compared with the paper media and the other is that efficiency of the advertisement is dynamically visualizing by link clicking. However, the Internet advertisement is passive advertisement for the companies, since the advertisement is appeared by relationship between the advertisement and some searching keywords. On the other hand, the paper media's advertisement is certainly appeared on the paper media. We suggest two problems of the paper media's advertisement. One is the advertisement fee does not consider the demand of the advertiser dynamically. This means that it is not easy to reflect some demand from the advertiser candidates to the advertisement fee, since the fee is decided by the owner of the paper media unilaterally. The other is that the paper media only focus on some especial readers. The Internet advertisement is available for unspecified number of people who visit such searching sites. On the other hand, the paper media can provide the advertisement to only their subscribers. Also it is difficult to run the advertisement on widely paper medias in the world for public relations activities. We create an insertion bidding system, called GlobalAd, for solving the second problem. GlobalAd system supports to the advertiser who tries to insert his advertisement to world paper medias. Also this system can insert the advertisement to multiple paper medias without individual negotiations. Actually, we have run an experiment which inserts the advertisement to the Korean newspaper by using the GlobalAd system. This paper proposes an advertiser determination method using auction system as dynamic advertisement fee decision mechanism considered a demand of the company for solving the first problem.

2 Paper Media Advertisement Scheme

Generally, a traditional advertisement scheme is that the advertiser offers the advertisement insertion to some domestic paper media. In case of offering to oversea, it needs to spend long time to insert the advertisement. To offer the advertisement to the paper media, the advertiser uses a phone, facsimile or e-mail. Existing advertisement's scheme requires too much costs to insert own advertisement on multiple paper medias. Because of this, the paper media's income of the advertisement is decreasing. Since the newspaper's advertisement also decreasing, the news paper company develops a new offering service model by using web based system like the Internet advertisement system. "Adstuff.com"[3] and "releasemyad.com"[4] are one of online newspaper's advertisement offering system. The advertiser can insert own advertisement on some national newspapers by using their system. However the user cannot insert the advertisement on other country's news paper. When the user uses this advertisement offering system, first of all, he/she choices some newspapers, dates and sizes of the advertisement. Second, the system offers an advertisement fee

to the user. If the user agrees this fee, then the system requires some materials of the advertisement to the user. The user uploads the materials to the system's database, and then pays the fee by using secured paying system, such as Paypal system. After that, the system send the advertisement to advertisement agency. This system have an expectation in which the advertisement's income is increasing, however, there are many problems to increase more advertisement's incomes, since this system cannot offer the advertisement to multiple countries.

3 GlobalAd System

In this section, we explain an application of our advertisement auction. In the Internet advertisement, it is important to calculate the number of clicks of the link of advertisement shown on the webpage. The number of successful trade after clicking an advertisement is also sometimes considered to know the quality of the advertisement. On the other hands, actual newspapers are not clicked by the subscribers, and business providers (who apply to publish their advertisement) may know the reputation of their advertisement from trading history using survey. Generally, because of the strong limitation of that, it is easier to make a formalization to determine winners in actual advertisement auction than the online advertisement auction, except for the constraints regarding space, multiple pages, position, and size.

There are some web-based advertisement application systems to be used in actual newspaper. None of them provide the procurement, winner-determination, and bidding mechanisms. The GlobalAd is a useful system to apply an advertisement that is used internationally. The GlobalAd is developed with a technology that will help a user to book its advertisement in any newspaper all over the world. Figure 3 shows the architecture of GlobalAd. First, a user will be able to select its preferences viz. its country, state, city, newspaper and date on which it wants to publish its advertisement in the selected newspaper. Upon selected preferences like size of the advertisement and newspaper, a bill is generated. If the user agrees to the amount in the bill, it is asked to upload its material on the database. After uploading the material, the user has to make payment via its credit/debit card or its bank account over a secured payment gateway of the Paypal. On making payment, a receipt is generated for the user. Also, an email of user preferences is sent to the advertisement agency on making the payment. On the basis of the preferences and the material uploaded by the user on the database, the advertisement agency books the advertisement in the newspaper selected by the user in its preferences. On the date selected by the user in its preferences, it can view the copy of the published advertisement in the newspaper by clicking a link on the Global Ad website. In this way, the Global Ad website will help a user book its advertisement in any newspaper of its choice all the world.

The GlobalAd currently provides the simple function explained above, however the combinations of users preferences are complicated; users have some preferences including advertisement's size, position, page, order, and cost. If users' some preferences are overlapped, it is rational to determine winners in the economics

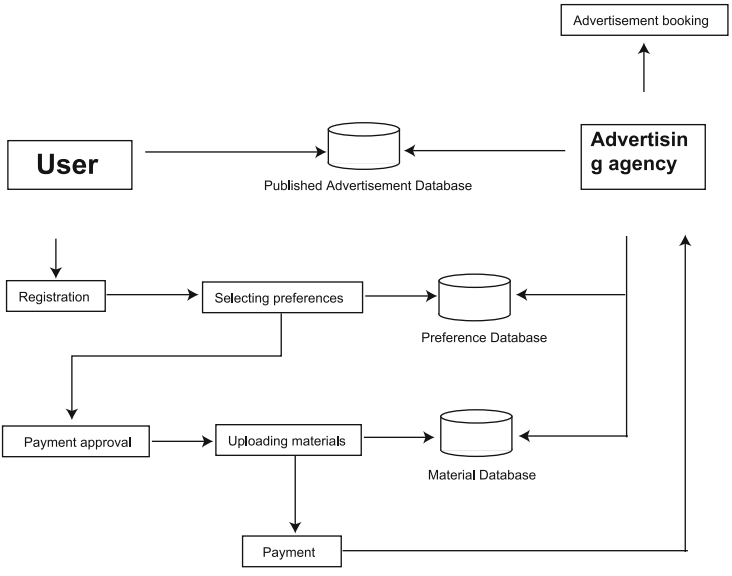


Fig. 1 System Architecture of GlobalAd

viewpoints. Namely, the auction type becomes applied multiple auctions or combinatorial auctions. Figure 3 shows the process using the auction-based winner determination in the GlobalAd. Using the second price auction, it is easy to determine winners, that becomes social surplus is maximum. The formalization is shown as follows. We show a multiple auction formalization. Suppose that b_{ij} is a bid value of advertiser i for a frame j. Also let N be a set of advertisers and F be a set of frames. Then,

$$\text{(AP)} \left| \begin{array}{ll} \text{maximize} & \sum_{i \in N} \sum_{j \in F} b_{ij} x_{ij} \\ \text{subject to} & \sum_{j \in F} x_{ij} \leq 1, \ \forall i \in N \\ & \sum_{i \in N} x_{ij} = 1, \ \forall j \in F \\ & x_{ij} \in \{0, 1\}, \ \forall i \in N, \ j \in F \end{array} \right.$$

x_{ij} is a binary decision variable. If x_{ij} takes 1, then the system allocates a frame j to an advertiser i. Let $\boldsymbol{X}^*$ be an optimal solution of the problem (AP). The optimal solution $\boldsymbol{X}^*$ shows the winners. The problem (AP) is one of general allocation problems, it is easy to compute by using some integer programming solver such as CPLEX[7] or Gurobi[8]. There are some paid softwares, however, the problem (AP) is also able to solvable by using graph theoretical technique written by [9] and [10].

However, this formulation is not applied to real system, because the real advertisement allocation problem to the newspaper includes some packing problem. In this paper we reformulate this auction based allocation mechanism as 2-dimensional packing problem.

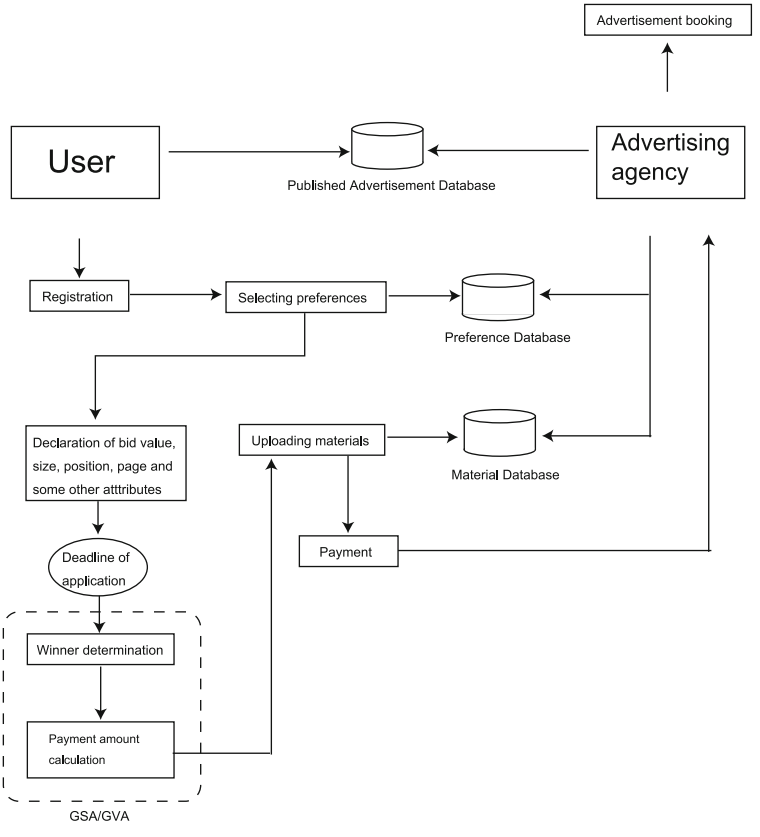


Fig. 2 Process of Auction-based Winner Determination

4 Advertisement Allocation Problem

GlobalAd system advances increasing advertisement's incomes, however, the inserted advertisement decision method in which multi advertisers compete in the trading, is not discussed. Table 1 and Fig. 4 shows an example of multi advertisers competition case. In table 1, when the size of the paper is that width is 380.0mm

Table 1 Example of advertisement auction. (width, height)

	space a	space b	space c	space d
bid. 1	\$40	\$10	\$50	\$100
bid. 2	\$20	\$50	\$30	\$80
bid. 3	\$30	\$20	\$35	\$90
bid. 4	\$0	\$10	\$80	\$75
bid. 5	\$20	\$60	\$110	\$200

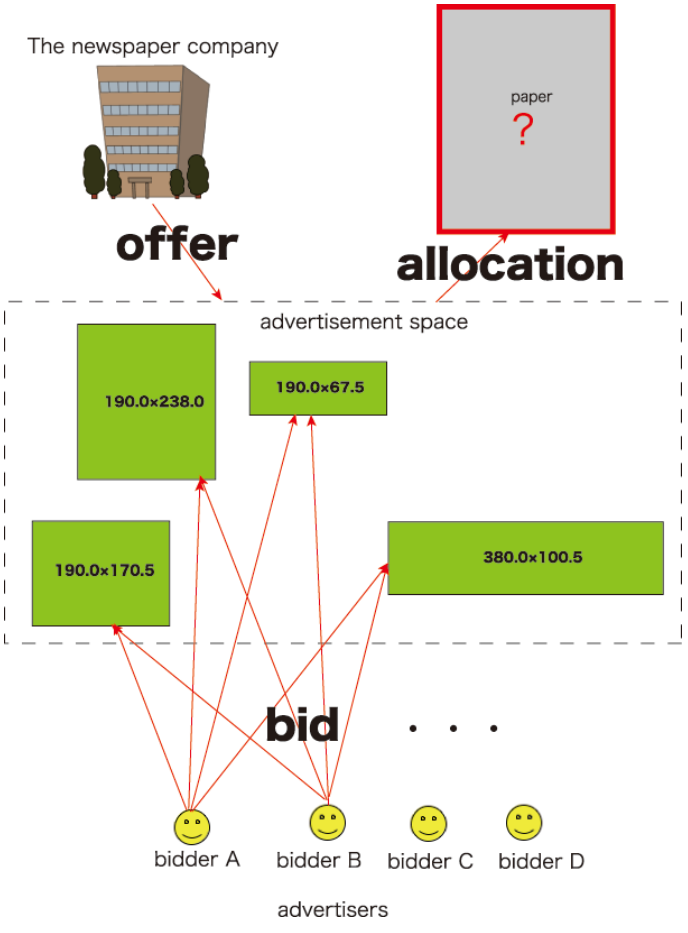


Fig. 3 Advertisement auction

and height is 540.0mm, each company bids to different four advertisement space offered by the owner. A space a, b, c and d has a size $(190.0, 67.5)$, $(190.0, 170.5)$, $(380.0, 100.5)$ and $(190.0, 238.0)$, respectively. In this case, the bidder 1, 2, 4 and 5 gets a space a, b, c and d, respectively. And the total value is \$370.

It is known the rectangle packing problem for the problem which put some rectangles on 2-dimension plane surface. This problem does not solve the auction's winner determination problem, since the problem consider only the size of rectangle instead the value of the rectangle.

4.1 Rectangle Packing Problems

Rectangle packing problems are the problems in which several sizes of rectangles are packed on the 2-dimension plane surface without overlapping each rectangle[12].

Given a base rectangle with height H and width W and a set of rectangles $N = \{1, 2, ..., n\}$, also given a height h_i and a width w_i for each rectangle i. Then we compute each rectangle's left lower point's coordinate (x_i, y_i) on the base rectangle under next two constraints[11].

Constraint 1 Each rectangle i does not protrude from the base rectangle:

$$0 \leq x_i \leq W - w_i, \ \forall i \in N \tag{1}$$
$$0 \leq y_i \leq H - h_i, \ \forall i \in N \tag{2}$$

Constraint 2: A rectangle pair (i, j) does not overlap. Hence, at least one condition holds from the follows:

$$x_i + w_i \leq x_j \tag{3}$$
$$x_j + w_j \leq x_i \tag{4}$$
$$y_i + h_i \leq y_j \tag{5}$$
$$y_j + h_j \leq y_i \tag{6}$$

The formula 3 shows a rectangle i is putted on the left side of a rectangle j. Similarly, the formula 4, 5 and 6 shows the rectangle i is putted on the rectangle j's right, under and top side, respectively.

4.2 Formulation of Advertisement Allocation Problem

The advertisement allocation problem cannot be represented by the simple rectangle packing problem, since the rectangle packing problem is not considered some rectangle's value. Hence we propose a combination the rectangle problem and a knapsack problem for representing the advertisement allocation problem.

We describe an advertisement allocation model. There are m spaces as selling items, and let $N = \{1, ..., n\}$ be a set of companies as bidders. Each space j has a height h_j and a width w_i. Also let $v_{ij} \in \Re_+ = \{x \in \Re \mid x \geq 0\}$ be an evaluation value for the space j from a bidder i. Then the allocation problem is following integer programming problem (IP).

$$(IP) \left| \begin{array}{ll} \text{maximize} & \sum_{i \in N} \sum_{j=1}^{m} v_{ij} z_{ij} \\ \text{subject to} & \sum_{j=1}^{m} z_{ij} \leq 1, \ \forall i \in N \\ & \sum_{i \in N} z_{ij} \leq 1, \ \forall j = 1, ..., m \\ & z_{ij} \in \{0, 1\}, \ \forall i \in N, \ j = 1, ..., m \end{array} \right.$$

z_{ij} is a binary decision variable. If z_{ij} takes 1, then the system allocates a apace j to an bidder i. First constraint shows that each bidder can get at most one space and second constraint shows that each space is allocated to at most one bidder. Z^* is an optimal solution of this problem, however, is not optimal solution of the advertisement allocation problem, since the integer programming problem does not

consider a packing of the space. Hence we should consider the following problem to solve a packing of advertisements.

2-dimension knapsack problem: For each i of given a set of item $I = \{1, ..., n\}$ has a height h_i, a width w_i and a value c_i. 2-dimension knapsack problem is that it maximizes a sum of item values under a rectangle packing's constraints.

$$\text{find } I^* \subseteq I, \text{ such that}$$
$$\max_{\tilde{I} \subseteq I} \left\{ \sum_{i \in \tilde{I}} c_i \middle| \begin{array}{l} 0 \leq x_i \leq W - w_i, \ \forall i \in \tilde{I} \\ 0 \leq y_i \leq H - h_i, \ \forall i \in \tilde{I} \\ \text{item } i \text{ and } j \text{ does not overlap} \end{array} \right\}$$

In this problem, there are $O(2^n)$ subsets of the set I. Moreover, for every subsets, we should solve a rectangle packing. Since it is known that a rectangle packing problem is one of NP-hard problems, we do not expect some exact solving algorithm. Also there are some cases that , for $I' \subseteq I$, the problem is infeasible. Hence, we should choice appropriate subsets of I, and evaluate a total value.

If we want to compute an optimal solution of the advertisement allocation problem, then we should compute the two problems at same time, however, it is impossible to solve the two problem at same time. Therefore we propose a 2-step algorithm for solving the advertisement allocation problem. First of all, we solve the (IP) by using something integer programming solver such as CPLEX or groubi. After that we create an instance of the 2-dimension knapsack problem by using the optimal solution of (IP). Finally, we solve the 2-dimension knapsack problem by using following algorithm.

Step 1. $I^{(0)} = \{i \in I \mid z^*_{ij} = 1, i \in N\}$. $\bar{W}$ and $\bar{H}$ are real decision valuable. Then we solve the following problem.

$$\begin{array}{ll} \text{minimize}(I^{(0)}) & \tilde{W}\tilde{H} \\ \text{subject to} & 0 \leq x_i \leq \tilde{W} - w_i, \ \forall i \in I^{(0)} \\ & 0 \leq y_i \leq \tilde{H} - h_i, \ \forall i \in I^{(0)} \\ & \text{(Constraint 2)}, \ \forall i \in I^{(0)} \end{array}$$

Step. 2 Let $(\bar{W}*, \bar{H}*)$ be a solution computed by Step. 1. Then we compute a divergency from given value (W, H). We create $I^{(1)} \subseteq I^{(0)}$ based on divergency value and item value.

Step. 3 We solve a minimization problem (minimize (I^1)).

Step. 4 Loop the Step. 2 and Step. 3 and make a feasible solution of original problem.

4.3 Advertisement Allocation Auction

We consider the following constraints for the advertisement allocation auction.

- It must not protrude the advertisements on the paper.
- It must not overlap each advertisement on the paper.

Fig. 4.3 shows an example of our proposing auction system. The newspaper company put a paper size and some spaces for advertisements. Then the bidder bids to some spaces, and the system decides an allocation and packing of the advertisement by solving the above problem.

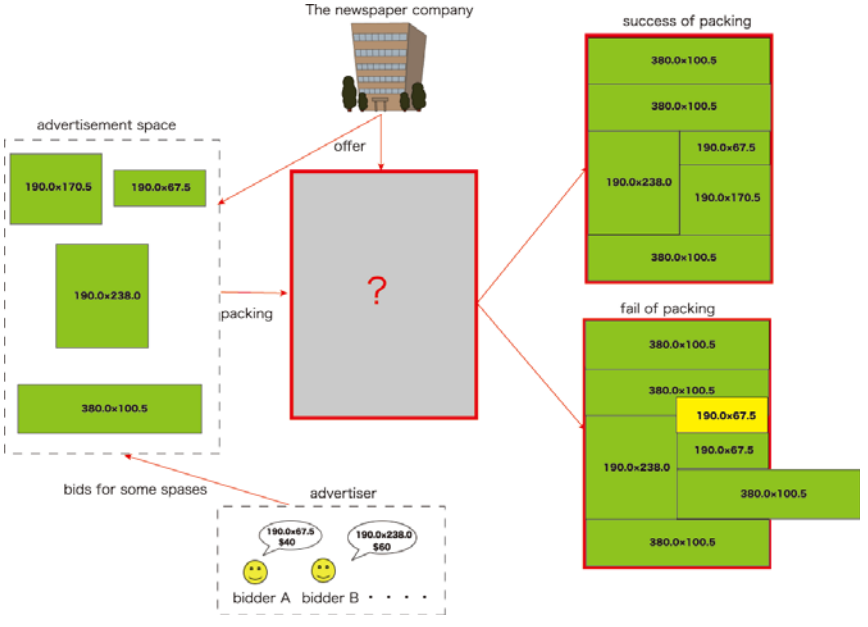


Fig. 4 Example of advertisement auction system

5 Conclusion

This paper focus on GlobalAd system for solving paper media's income problem. However in GlobalAd system, it does not discuss when there are many advertisers in the system. This paper propose a model of the advertisement auction by using 2-dimension knapsack problem formulation as optimal advertisement location problem. Our future work is to create an auction mechanism based on second price auction which has very useful properties[5], also, to apply this method to the GlobalAd system.

Acknowledgements. IThis research is supported by the foundation "Hattori-Hokokai".

References

1. http://www.yahoo.com
2. http://www.google.com
3. http://www.adstuff.com/
4. http://www.releasemyad.com/
5. Krishna, V.: Auction Theory. Academic Press (2002)
6. Edelman, B., Ostrovsky, M., Schwarz, M.: Internet Advertising and the Generalized Second Price Auction: Selling Billions of Dollars Worth of Keywords. American Economic Review 9(1), 242–259 (2007)
7. http://www-01.ibm.com/software/integration/optimization/cplex-optimizer
8. http://www.gurobi.com/
9. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms, and Applications. Prentice-Hall (1993)
10. Korte, B.H., Vygen, J.: Combinatorial optimization: theory and algorithms. Springer (2004)
11. Korf, R.E.: Optimal Rectangle Packing: New Results. In: Proceeding of the 14th International Conference on Automated Planning & Scheduling, pp. 142–149 (2004)
12. Backer, B.S., Coffman Jr., E.G., Rivest, R.L.: Orthogonal packing in two dimensions. SIAM Journal on Computing 9, 846–855 (1980)

Effects of Utility Functions on Network Response Time and Optimization

Chris Johns, Kevin Mak, Gongzhu Hu, and Wenying Feng

Abstract. Network optimization is a classic and effective approach for allocating network resources in such a way that certain measure of the network utilization is optimized. Network models and algorithms have been proposed and developed for solving the optimization problems. However, we haven't seen studies on the effect of the utility functions on the network response time when the overall utilization of the network is maximized. In this paper, we investigate this problem with simulation experiments on a simple 4-node network using two different utility functions, a logarithmic function and a linear function. We fine tune the network transmission rates near their optimal values on several routes and observe the network response time. Our preliminary study showed that different utility functions do have impact on the response time on individual routes.

Keywords: network congestion, network optimization, utility function, network model.

1 Introduction

As computer networks, particularly the Internet, became an essential tool for communications in the last 20-30 years, it is crucial for the networks to

Chris Johns · Kevin Mak · Wenying Feng
Departments of Computing & Information Systems and Mathematics,
Trent University, Peterborough, Ontario, Canada, K9J 7B8
e-mail: {christopherjohns,kevinmak,wfeng}@trentu.ca

Gongzhu Hu
Department of Computer Science, Central Michigan University,
Mt. Pleasant, MI 48859, USA
e-mail: hu1g@cmich.edu

R. Lee (Ed.): Software and Network Engineering, SCI 413, pp. 77–88.
springerlink.com

be stable to provide effective communication services. That is, the networks should deliver data to their intended destinations as expected in a timely manner. One big problem that had affected the behavior of computer networks is *congestion*.

The network congestion problem was first addressed by researchers in the 1980's when the Internet started to become "crowded". Jacobson proposed a congestion control algorithm [4] that was later adopted in the Transmission Control Protocol (TCP). Chiu and Jain [3] studied this problem from a view of appropriate allocating network resources (bandwidth, for example) among competing users. The resource allocation concept was generalized by Frank Kelly who introduced a network model and presented a network optimization method [5] to address the congestion problem. In his approach, the problem was posed as an optimization problem to maximize the overall network utility subject to certain constraints. A charging scheme with usage-based pricing and various fairness criteria were used to formulate the constraints.

Since then, various solutions to the optimization problem have been proposed including using differential equations, queueing theory, and discrete optimization, among others. The differential equation based solution in [6] was utilized in rate control algorithms that were compared in terms of stability and fairness. In the queueing theory approach presented in [10], the network utility was optimized by a sequence of queueing networks. A greedy primal-dual algorithm was developed in [9] to maximize queueing network utility for congestion control. For discrete optimization, an approach to address data uncertainty was given in [2] that allows control of the tradeoff between cost and robustness in addition to optimization. A graph theory based general network optimization approach was presented in [1]. The recent monograph [8] provided a comprehensive review and analysis of the network optimization problem.

These solutions, however, have considered utility functions in a general terms aiming to maximize the overall utilization of network resources. Few have studied and compared the effects of different utility functions on the performance on the network while being optimized. In this paper, we consider two specific utility functions and compare their effects on the network response time with the data flow rates at or near their optimal values. The optimal rates are obtained using the Kelly's optimization model. The utility functions we considered include a logarithmic function and linear function, which were used in our simulation experiments on a simple network of four nodes interconnected by four links. First, we solve the 4-node network optimal transmission rates mathematically with these utility functions, then we simulate these solutions and compare the output results in terms of network response time. The simulation results show that the response time with the logarithmic function is uniform among the four routes, while the linear utility function produced faster response time on some routes.

2 The Optimization Model

We shall first describe the optimization model introduced by Kelly, et al [6] as our study is based on this model. A network consists of the following:

S: a set of traffic sources (nodes) ;
J: a set of links connecting the nodes;
C: a set of capacities, $c_j \in C$ is associated with link $l_j \in J$;
R: a set of routes, where $r \subset J$ for each $r \in R$.

Since each source $s \in S$ is associated with a route $r \in R$ to transmit traffic, we use the same index r for a route and the source that transmits traffic along route r. Note that a route r consists of links, i.e. $r \subset J$, we can represent the routing scenario of a network as a matrix $\boldsymbol{A}$ with the elements defined as:

$$\boldsymbol{a}_{jr} = \begin{cases} 1 & \text{if } l_j \in r, \\ 0 & \text{otherwise.} \end{cases}$$

That is, the matrix element $\boldsymbol{a}_{jr}$ indicates whether link l_j is on route r. In addition, a route r is associated with a transmission rate x_r. Since a route r is used by a user at a traffic source to transmit data, we can also use r to denote the user. The network utilization of a user r is measured by a utility function, denoted as $U_r(x_r)$. The utility functions are normally assumed to be continuously differentiable over the range $x_r \geq 0$ and that $U_r(0) = 0$ for every r. Since utilities are additive and we want to find the optimal rates x_r that maximizes the total value of utilities $\sum_{r \in R} U_r(x_r)$ provided that transmission rates are non-negative and do not exceed the capacities of the links. Hence the optimization problem is presented as the following system:

SYSTEM$(U, \boldsymbol{A}, \boldsymbol{C})$:

$$\max \sum_{r \in R} U_r(x_r)$$

subject to

$$\boldsymbol{Ax} \leq \boldsymbol{C}$$

over

$$\boldsymbol{x} \geq \boldsymbol{0}.$$

where $\boldsymbol{x} = (x_1, \cdots, x_n)^T$ is a vector of the rates on the n routes and $\boldsymbol{C} = (c_1, \cdots, c_n)^T$ is the link capacity vector.

We can solve this optimization problem using the method of Lagrangian multipliers and the Karush-Kuhn-Tucker (KKT) theorem [7]. The Lagrangian of the function to be maximized is

$$L = \sum_{r \in R} U_r(x_r) - \sum_{j \in J} \mu_j \left(\sum_{r : j \in r} x_r - c_j \right).$$

The KKT condition (for each x_r) is

$$\frac{\partial L}{\partial x_r} = U_r'(x_r) - \lambda_r = 0, \tag{1}$$

where μ_r and λ_r are the Lagrangian multipliers, and these conditions hold:

$$\lambda_r = \sum_{j:j\in r} \mu_j, \tag{2}$$

$$\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{C}, \tag{3}$$

$$\boldsymbol{x} \geq \boldsymbol{0}, \tag{4}$$

$$\mu_j \geq 0, \tag{5}$$

$$\mu_j \left(\sum_{r:j\in r} x_r - c_j \right) = 0. \tag{6}$$

3 Analysis Results

For the analysis, we use a simple network example of four nodes and four users shown in Fig. 1 to illustrate the process of solving the optimization problem.

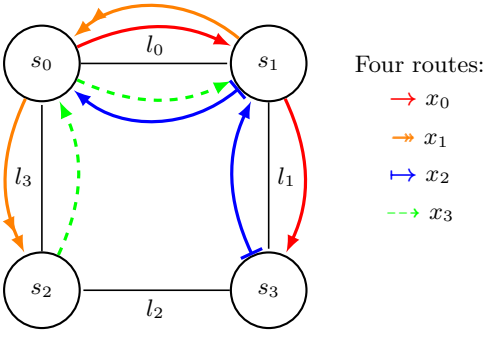


Fig. 1 Four routes on a 4-node network

In this figure, the four traffic sources s_i are connected by the links l_j, $i, j \in \{0, 1, 2, 3\}$ and the color coded arrows represent the four routes $r = 0, 1, 2, 3$ with transmission rates x_r. For example, both routes 0 (red) and 2 (blue) are composed of the two links l_0 and l_1, while both routes 1 (orange) 3 (green) are composed of links l_0 and l_3. The link l_2 is not used by any route. Hence, the route-link matrix of this network is represented as

$$\boldsymbol{A} = \begin{pmatrix} 1\ 1\ 1\ 1 \\ 1\ 0\ 1\ 0 \\ 0\ 0\ 0\ 0 \\ 0\ 1\ 0\ 1 \end{pmatrix}.$$

Assuming all the links have capacity 1, we have

$$\boldsymbol{C} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

From equation (2) of the KKT condition, the λ_i are calculated as

$$\begin{aligned} \lambda_0 &= \mu_0 + \mu_1, \\ \lambda_1 &= \mu_0 + \mu_3, \\ \lambda_2 &= \mu_0 + \mu_1, \\ \lambda_3 &= \mu_0 + \mu_3. \end{aligned}$$

Applying equation (6) with the transmission rates $\boldsymbol{x} = (x_0, x_1, x_2, x_3)^T$, we get the following equations:

$$\begin{aligned} \mu_0 \left(x_0 + x_1 + x_2 + x_3 - 1\right) &= 0, \\ \mu_1 \left(x_0 + x_2 - 1\right) &= 0, \\ \mu_2 \left(0 - 1\right) &= 0, \\ \mu_3 \left(x_1 + x_3 - 1\right) &= 0. \end{aligned}$$

To solve for $\boldsymbol{x}$, we need to use the KKT condition (1) to apply to the utility function U with respect to each x_r. In the following, we consider two different utility functions to analyze and compare their effects on the optimization results. The functions we use include a logarithmic function (U_0) and a linear function (U_1). These functions abide by the assumptions of being continuously differentiable and $U(0) = 0$. As well, all utility functions abide by the KKT condition requirements.

3.1 Utility Functions

The two utility functions, U_0 and U_1, used in our analysis are given below.

Logarithmic utility function. The logarithmic function is

$$U_0 = a \ln\left(x + 1\right),$$

where a is a constant. For this function, the KKT condition (1) is

$$\frac{\partial L}{\partial x_r} = \frac{a}{x_r + 1} - \lambda_r = 0. \tag{7}$$

Linear utility function. The linear function is

$$U_1 = bx,$$

where b is a constant. For this function, the KKT condition (1) is

$$\frac{\partial L}{\partial x_r} = b - \lambda_r = 0. \tag{8}$$

3.2 Analysis Solutions

For the logarithmic utility function U_0, we set $a = 1$ and solve the equation (7). The results of the x_i's are all equal: $x_1 = x_2 = x_3 = x_4 = 0.25$.

For the linear utility function U_1, we set $b = 1$ and the solution of equation (8) is not unique as long as the following condition is met:

$$x_0 + x_1 + x_2 + x_3 = 1, \tag{9}$$

where $0 \leq x_r \leq 1$ for all r. For example we can have

$$x_0 + x_1 + x_3 = 1,\ x_2 = 0$$

or

$$x_0 = x_3 = 0.125,\ x_2 = 0.5,\ x_3 = 0.25.$$

The second example solution is the one we used for the first simulation experiment for U_1.

4 Simulation

Simulation experiments were conducted on a 4-node network shown in Fig. 1 using the tool NS-3 (Network Simulator 3), which is a discrete event network simulator. We set up point-to-point connections as links between two nodes. Users echo UDP packets from a client node to a server node using the appropriate connections as a measure of the time spent on the transmission of data. The goal of the simulation was to see what rates the users use to transmit the data along their routes would achieve the optimal utilization of the network as measured by the utility functions, and what difference the different utility functions would make on the network performance in terms of the response time. Various rates were used by adjusting the payload (amount of actual data transmitted, in bits) for the UDP packets.

4.1 Results for Analysis Solutions

For the purpose of comparison of the effects of the utility functions on the network performance, we use several different values for the payload sizes on the routes to adjust their traffic rates. The distribution of the payloads are based on the analytical solutions obtained in Section 3.2. The link capacity was set to 1024 bps. Since all the four routes using U_0 have the same rates, we divide the link capacity evenly among these routes. Similarly, the capacity is distributed according to the analytical results for U_1. With these rates, we recorded the response time of each route that is the elapsed time between the packet was sent to the time an echo from the client was received by the host. The results are shown in Table 1.

Table 1 Response time with different payload sizes using U_0 and U_1

route	log function U_0		linear function U_1	
	payload (in bits)	resp time (in seconds)	payload (in bits)	resp time (in seconds)
x_0	256	0.277	128	0.333
x_1	256	0.277	256	0.277
x_2	256	0.277	512	0.518
x_3	256	0.277	128	0.335

The results show that U_0 produced a little better response time than U_1. And, since all users have the same utility, it can also be said that U_0 generated more "fair" rates when no other factors (such as cost) were considered.

4.2 Effects of Adjusted Rates

To find how changes in the rates would affect the network performance, we adjusted the rates on the routes in two ways. One is to to make the rates deviate from their optimal values (analytical results) slightly, and the other is to change by a large amount. The rates were changed by increasing or decreasing the sizes of the payload.

4.2.1 Small Changes of Payload Size

First, we modified the rates by changing the payload size slightly ($\pm$ 5 bps). We ran two cases, one was to decrease the payload sizes for x_0, x_1 and increase the sizes for x_2, x_3. The results are given in Tables 2(a) and 2(b), respectively. Note that the changes were made according to the analytical solution given in Equation (9). That is, $\sum_{r=0}^{3} x_r = 1024$ bps, the link capacity.

Table 2 Response time with small changes of payload sizes

(a) $(x_0, x_1) - 5$ bps; $(x_2, x_3) + 5$ bps

route	log function U_0		linear function U_1	
	payload (in bits)	resp time (in seconds)	payload (in bits)	resp time (in seconds)
x_0	251	0.278	123	0.336
x_1	251	0.273	251	0.272
x_2	261	0.282	517	0.523
x_3	261	0.282	133	0.340

(b) $(x_0, x_1) + 5$ bps; $(x_2, x_3) - 5$ bps

route	log function U_0		linear function U_1	
	payload (in bits)	resp time (in seconds)	payload (in bits)	resp time (in seconds)
x_0	261	0.282	133	0.331
x_1	261	0.282	261	0.282
x_2	251	0.273	507	0.513
x_3	251	0.272	123	0.331

4.2.2 Large Changes of Payload Size

Reversely, we experimented with large change of the payload sizes. The results are shown in Tables 3(a) and 3(b). Again, the sum of the x_rs was kept at the link capacity (1024 bps) for all the changes.

The response times with different payload sizes are illustrated visually in Fig. 2 for U_0 and Fig. 3 for U_1, respectively. It is seen that the plots are very close for all the routes using U_0 as for this utility function the optimal rates were calculated to be the same value.

For U_1, the changes of response time are quite different. Notice that the response time is pretty much flat for routes x_0 and x_1 when payload size is small, even decreased a bit for x_0 when the payload increases in the range of (120–200).

4.2.3 Rate of Change of Response Time

From the above figures, we see that changes to the payload sizes do have impact on the response time. The rate of changes were calculated and shown in Table 4, where the result for U_0 is in 4(a) and that for U_1 is given in 4(b).

The relationship between the change (% increase or decrease) of response time and the payload sizes for U_0 is visually shown in Fig. 4. We see that

Table 3 Response time with large changes of payload sizes

(a) $(x_0, x_1)\,/\,2$; (x_2, x_3) increase

route	log function U_0		linear function U_1	
	payload (in bits)	resp time (in seconds)	payload (in bits)	resp time (in seconds)
x_0	128	0.236	64	0.326
x_1	128	0.271	128	0.271
x_2	384	0.398	608	0.608
x_3	384	0.433	224	0.448

(b) (x_0, x_1) increase; $(x_2, x_3)\,/\,2$

route	log function U_0		linear function U_1	
	payload (in bits)	resp time (in seconds)	payload (in bits)	resp time (in seconds)
x_0	384	0.398	192	0.281
x_1	384	0.398	384	0.398
x_2	128	0.236	416	0.428
x_3	128	0.236	32	0.245

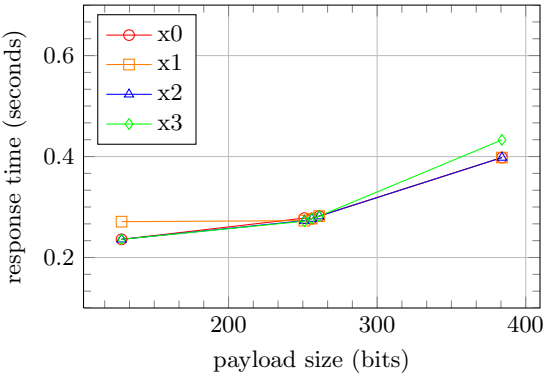


Fig. 2 Response time vs. payload size for U_0

the changes of the reponse time have a lower slope before the payload sizes reach their optimal values (256 bps for all the routes) and then increase much faster (steep slope), with x_3 having the most steep slope and x_1 is the most flat one.

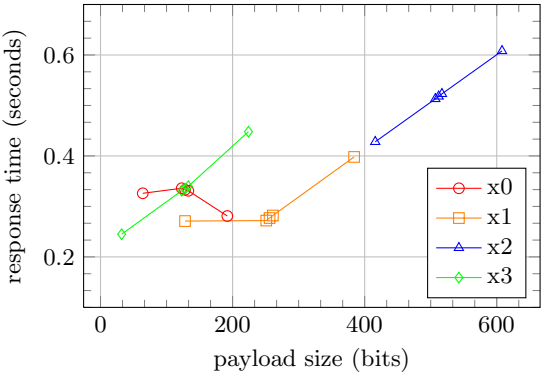


Fig. 3 Response time vs. payload size for U_1

Table 4 Rate of change in response time with changes of payload sizes

(a) for U_0

route	change in response time (in %)			
	$(x_0, x_1) \downarrow$, $(x_2, x_3) \uparrow$		$(x_0, x_1) \uparrow$, $(x_2, x_3) \downarrow$	
	small	large	small	large
x_0	0.36	−14.80	1.81	43.68
x_1	−1.44	−2.17	1.81	43.68
x_2	1.81	43.68	−1.44	−14.80
x_3	1.81	56.32	−1.81	−14.80

(b) for U_1

route	change in response time (in %)			
	$(x_0, x_1) \downarrow$, $(x_2, x_3) \uparrow$		$(x_0, x_1) \uparrow$, $(x_2, x_3) \downarrow$	
	small	large	small	large
x_0	0.90	−2.10	−6.61	−15.62
x_1	−1.81	−2.17	1.81	43.68
x_2	0.97	17.37	−0.97	−17.37
x_3	1.49	33.73	−1.19	−26.87

Fig. 5 illustrates the relationship between the rate of change of the response time and the payload size for U_1.

It is seen that routes x_3 has steep slope in the plot meaning that the same change of payload size caused more rapid change in the response time on the route, while the route x_2 has the most slow slope except for x_1 when

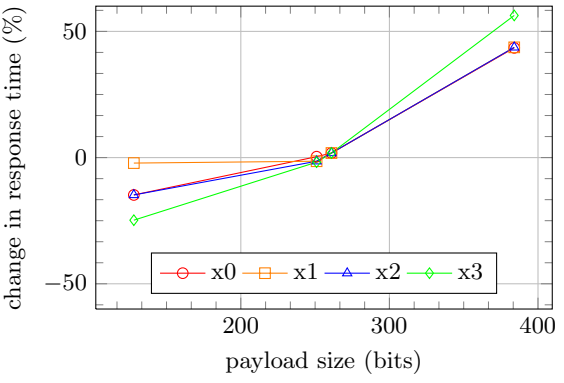


Fig. 4 Change in response time vs. payload size for U_0

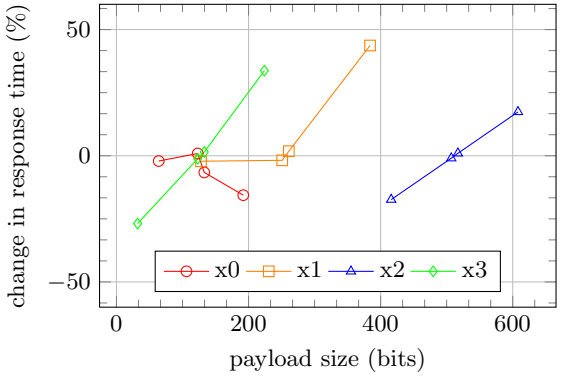


Fig. 5 Change in response time vs. payload size for U_1

payload size is smaller than 220 bps where the slope for x_1 is flat. These results are no surprise simply because routes x_3 was allocated the smallest payload (hence highest transmission rate) while the payload on x_2 was much larger. It is interesting to note that the plot for x_0 is not a monotonic function of payload due to the response time resulted from the simulation that was shown in Fig. 3.

5 Conclusions

Network optimization as a solution to the congestion problem is well studied and various approaches have been proposed. Most previous work on network optimization considered maximizing the overall utilization of the network with generalized utility functions. Few have looked at the impact of different

utility functions on the network response time on individual routes while performing optimization to find the best data transmission rates. In this paper, we studied the Kelly's network optimization model with two different utility functions, and investigated how the network response time is related to the changes of the data transmission rates allocated to the routes. Simulation results, although very preliminary, show that the logarithmic utility function generated more uniform optimal rates for the routes than the linear function. The response time changes somewhat proportionally to the changes to the rates.

We are currently conducting experiments with larger network with 100+ nodes and more utility functions with randomly generated link capacities (within a reasonable range, of course). We plan to apply per-user-based utility functions and other optimization approaches in the simulation to compare with the results from the Kelly's model.

Ackmowledgment. This project was partly supported by a grant from the NSRC (Natural Sciences Research Committee) of Trent University, Canada.

References

1. Alon, N., Awerbuch, B., Azar, Y., Buchbinder, N., Naor, J.S.: A general approach to online network optimization problems. ACM Transactions on Algorithms 2(4), 640–660 (2006)
2. Bertsimas, D., Sim, M.: Robust discrete optimization and network flows. Mathematical Programming 98(1), 49–71 (2003)
3. Chiu, D.M., Jain, R.: Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. Computer Networks ISDN Systems 17(1), 1–14 (1989)
4. Jacobson, V., Karels, M.J.: Congestion avoidance and control. ACM Computer Communication Review 18(4), 314–329 (1988)
5. Kelly, F.: Charging and rate control for elastic traffic. European Transactions on Telecommunications 8, 33–37 (1997)
6. Kelly, F., Maulloo, A.K., Tan, D.: Rate control in communication networks: shadow prices, proportional fairness and stability. Journal of the Operational Research Society 49(3), 237–252 (1998)
7. Kuhn, H.W., Tucker, A.W.: Nonlinear programming. In: Second Berkeley Symposium on Mathematical Statistics and Probability, pp. 481–492. University of California Press (1951)
8. Shakkottai, S., Srikant, R.: Network optimization and control. Foundations and Trends in Networking 2(3), 271–379 (2008)
9. Stolyar, A.L.: Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. Queueing Systems 50, 401–457 (2005)
10. Walton, N.S.: Utility optimization in congested queuing networks. Journal of Applied Probability 48(1), 68–89 (2011)

Features Detection from Industrial Noisy 3D CT Data for Reverse Engineering

Thi-Chau Ma, Chang-soo Park, Kittichai Suthunyatanakit,
Min-jae Oh, Tae-wan Kim, and Myung-joo Kang

Abstract. To detect features are significantly important for reconstructing a model in reverse engineering. In general, it is too difficult to find the features from the original industrial 3D CT data because the data have many noises. So it is necessary to reduce the noises for detecting features. This paper proposes a new method for detecting corner features and edge features from noisy 3D CT scanned data. First, we applied the level set method[18] to CT scanned image in order to segment the data. Next, in order to reduce noises, we exploited nonlocal means method[19] to the segmented surface. This helps to detect the edges and corners more accurately. Finally, corners and sharp edges are detected and extracted from the boundary of the shape. The corners are detected based on Sobel-like mask convolution processing with a marching cube. The sharp edges are detected based on Canny-like mask convolution with SUSAN method[13], which is for noises removal. In the paper, the result of detecting both features is presented.

1 Introduction

Features are significantly used as design elements to reconstruct an object model in reverse engineering. Especially in case of reconstructing a B-spline model from computed tomography (CT) scanned data, we need a curve

Thi-Chau Ma · Kittichai Suthunyatanakit · Min-jae Oh · Tae-wan Kim
Department of Naval Architecture and Ocean Engineering,
Seoul National University, Seoul, Korea
e-mail: ma.thi.chau@gmail.com, skittichai@hotmail.com
e-mail: {mjoh80,taewan}@snu.ac.kr

Chang-soo Park · Myung-joo Kang
Department of Mathematical Sciences, Seoul National University,
Seoul, Korea
e-mail: {winspark,mkang}@snu.ac.kr

R. Lee (Ed.): Software and Network Engineering, SCI 413, pp. 89–101.
springerlink.com

network including corners, sharp edges, ridges, etc. Fig. 1 shows a reverse engineering process from CT to B-spline. Feature detection has attracted lots of attention [1], [4], [5], [6], [7], [8], [9], [12]. However, the existing methods either depend on mesh generation or take time to check every voxel data. In this paper we propose a method how to detect those features, i.e. corners and sharp edges, from CT scanned data. First, we apply the level set method(LSM) to CT scanned image in order to segment the data. Using the segmented data, we can focus on not the whole volume but the surface in the process of detecting features. Next, in order to reduce noises, we exploit nonlocal means method to the segmented surface. This helps to detect the edge and corners more accurately. Finally, corners and sharp edges are detected and extracted from the boundary of the shape. The corners are detected based on Sobel-like mask convolution processing with a marching cube. The sharp edges are extracted based on Canny-like mask convolution with SUSAN method, which is useful to remove the remaining noises.

The paper is organized as follows. Section 2 shows related works and backgrounds. Section 3 presents our method how to detect corner and sharp edge features. Experiments and results are provided in section 4.

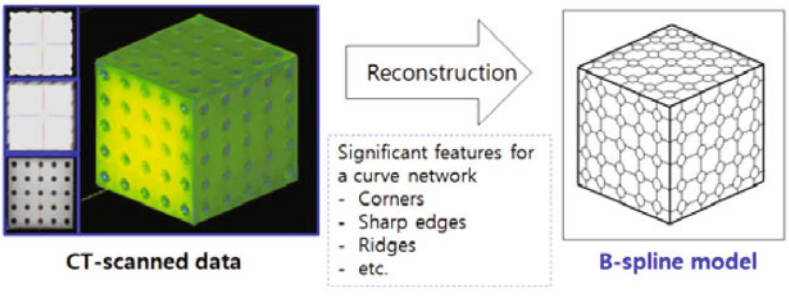


Fig. 1 Reverse engineering from CT scanned data to B-spline model

2 Related Works and Backgrounds

2.1 Related Works

Feature detection methods can be classified into two groups: polygon-based methods and point-based methods. There exist many techniques for feature detection and extraction relying on polygonal meshes [1], [5], [7], [9]. These techniques often consist of two steps: mesh generation and feature detection. In [7], [9], discrete extremities to extract feature lines are used in regular triangles. Singular triangles are treated specially based on neighbors of singular or regular triangles. In [1], a normal-based classification operator was used to classify edges into different types by combining some of thresholds.

In [5], Watanabe and Belyaev estimated the principle curvature extrema on dense triangle mesh. Then, these curvature extrema were used to form focal surface. The properties of focal surface were used to identify features.

Point based methods [4], [6], [8], [12] are more interesting because of the lack of knowledge concerning normal and connectivity information. Gumhold et al. [12] considered the k-nearest neighbors for every data voxel. They used Principal Component Analysis (PCA) to analyze the neighbors of each point. Eigenvalues and eigenvectors of coefficient matrix were used to determine the probability that the voxel belong to feature lines, borders or corner points. Pauly et al. [8] extended the PCA approach to multiple neighborhood sizes. The algorithm recognized all kinds of features. In [4] and [6], Gauss map was built for each data voxel. Voxels are classified by clustering Gauss map. In [11], 3D filters were extended from 2D edge detection filters.

Those mesh-based methods require mesh generation and thus need relying on the accuracy of mesh generation; while those point-based methods require procedure running every data voxel and thus have high computational cost. For our method, we use LSM and mask convolution on voxels. The method can be applied to detect feature without normal and connectivity information, like mesh-based. In addition, the computational time is reduced.

2.2 Segmentation Using Level Set Method

The level set method is very popular in computational fluid dynamics, computer graphics, image processing because of its advantage of handling the complicated topology and implementing easily. It represents the contour or the surface as the zero level set of a higher dimensional signed distance function. We can explain the detail as follows. Let the region $\Omega(t)$ be enclosed by the closed surface $\Gamma(t)$. Then the level set method uses the level set function $\phi(\mathbf{x}, t)$ to represent $\Gamma(t)$ as the zero level set of $\phi(\mathbf{x}, t)$ as Fig. 2. i.e.

$$\begin{aligned}
\phi(\mathbf{x}, t) &> 0 \text{ in } \Omega(t), \\
\phi(\mathbf{x}, t) &= 0 \text{ in } \Gamma(t), \\
\phi(\mathbf{x}, t) &< 0 \text{ in } \bar{\Omega}^c(t).
\end{aligned}$$

In the level set method, we discretize the domain into rectangular grids and have the value of $\phi(\mathbf{x}, t)$ at each grid. As we solve some partial differential equation using finite difference method, we can evolve the surface $\Gamma(t)$.

We employ the segmentation model using the level set method. In image processing, the mathematical model of segmentation is introduced by Mumford and Shah [16]. For a given image, they decomposed the image into piecewise-smooth approximation by minimizing Equation (1):

$$F(\mu, C) = \mu L(C) + \lambda \int_{\Omega} (u_0 - u)^2 d\Omega + \int_{\Omega/C} |\nabla u|^2 d\Omega, \tag{1}$$

where $L(C)$ denotes the length of C, which is the boundary of object to be detected, $\lambda > 0$ and $\mu > 0$ are parameters. Usually in the boundary of C, the intensity of the image u_0 changes rapidly. The first term makes the length of C as short as possible, that is, plays the role in reducing noises. The second term fits the image u to u_0 closely. The last term makes the image except edges as smooth as it can. This term also helps to denoising.

Chan and Vese[17] proposed a new model based on the above functional but without calculating the gradient. They assume that a given image is composed of two regions of approximately piecewise-constant intensities. They decompose the image into two regions using Equation (2).

$$\begin{aligned} F(c_1, c_2, C) = \mu \cdot Length(C) + \nu \cdot Area(inside(C)) \\ + \lambda_1 \int_{inside(C)} |u_0(x, y) - c_1|^2 dxdy \\ + \lambda_2 \int_{inside(C)} |u_0(x, y) - c_2|^2 dxdy, \end{aligned} \tag{2}$$

where $\mu \geq 0, \nu \geq 0, \lambda_1 > 0, \lambda_2 > 0$ are fixed parameters and the constants c_1, c_2 are the averages of u_0 inside and outside of C, respectively. The first and second term are regularizing terms to remove noises. The third and fourth term contribute to fitting of curve C in the following way. If the curve C is outside the object, the third term is positive and the fourth term is nearly close to zero. On the other hand, when the curve C is inside the object, the third term is almost zero and the fourth term is positive. If C overlaps with the object partially, both the third and the fourth term are positive. In the end, the fitting terms are minimized if C is on the boundary of the object.

Song and Chan[2] considered the following three terms from the above Chan-Vese model:

$$F(c_1, c_2, C) = \mu \cdot Legnth(C) + \lambda_1 \int_{inside(C)} |u_0 - c_1|^2 + \lambda_2 \int_{inside(C)} |u_0 - c_2|^2 \tag{3}$$

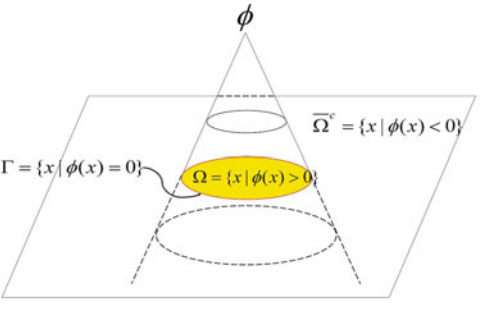


Fig. 2 Level Set Function

They can handle noises with only the length term because both the length term and the area term play the role in denoising. In order to apply the level set method to the above functional, they replace the curve C with a Lipschitz function ϕ. Then the previous functional can be modified as follows.

$$F(H(\phi), c_1, c_2) = \mu(\int_{\Omega} |\nabla H(\phi))|) + \lambda_1 \int_{inside(C)} |u_0 - c_1|^2 H(\phi) dx + \lambda_2 \int_{inside(C)} |u_0 - c_2|^2 (1 - H(\phi)) dx \quad (4)$$

where $H(\phi)$ is the Heaviside function of ϕ. They approximate $\int |\nabla H(\phi)| dx$ with

$$\sum_{i,j} \sqrt{(H(\phi_{i+1,j}) - H(\phi_{i,j}))^2 + (H(\phi_{i,j+1}) - H(\phi_{i,j}))^2} \quad (5)$$

where $\phi_{i,j}$ is the value of ϕ at the i, j-th pixel. In order to minimize the energy F, they propose the algorithm as follows.

First, set the initial value for ϕ. For example, we can set $\phi = 1$ for a half rectangular part in the image and $\phi = -1$ for the other part. Second, Let x be the value of current pixel and c_1, c_2 be averages for $\phi = 1$ and $\phi = -1$, respectively. Denote m and n as the number of pixels for $\phi = 1$ and $\phi = -1$, respectively. When $\phi(x) = 1$, compute the difference between the new and old energy:

$$\Delta F_{12} = (x - c_2)^2 \frac{n}{n+1} - (x - c_1)^2 \frac{m}{m-1}. \quad (6)$$

If $\Delta F_{12} < 0$, change $\phi(x)$ from 1 to -1 while if $\Delta F_{12} > 0$, remain $\phi(x)$ unchanged. Similarly when $\phi(x) = -1$, we compute

$$\Delta F_{21} = (x - c_1)^2 \frac{m}{m+1} - (x - c_2)^2 \frac{n}{n-1}. \quad (7)$$

If $\Delta F_{21} < 0$, change $\phi(x)$ from 1 to -1 and if $\Delta F_{21} > 0$, remain $\phi(x)$ unchanged. If denoising is needed, include the length term in the above approximation. Repeat the second step until the total energy F does not change.

This algorithm has the advantage of being very fast and being able to handling noises. After segmenting the 3D image directly with the above algorithm, we can get the binary image of which value inside the object is -1 and value outside the object is 1.

The computational time of this segmentation algorithm is $O(N)$, where N is the number of input data.

2.3 Nonlocal Means Filtering

In order to reduce noises in the segmented surface, we employ nonlocal surface restoration method addressed in [19]. In [19], they calculate the following gradient flow for surface denoising.

$$u_t = \Delta_w u := \frac{1}{2}\mathrm{div}_w(\nabla_w u) = \int_\Omega (u(y) - u(x))w(x,y)dy, \tag{8}$$

where weight $w(x,y)$ and similarity function $D(x,y)$ are defined by

$$w(x,y) = e^{-|x-y|^2/c_1} e^{-D(x,y)/c_2},$$
$$D(x,y) = \|\phi[x] - \phi[y]\|_2^2, x \in \Sigma_\delta, y \in N_x,$$

and N_x is a neighborhood of x within Σ_δ and $\phi[x]$ is a 3D patch of ϕ centered at x.

The strategy is as follows. First, it begins a surface which is the zero level set of a signed distance function ϕ. Then, calculate the following discretized version of the equation (8), iteratively.

$$\phi_j^{k+1} = \phi_j^k + dt \sum_{l \in N_j} w_{jl}(\phi_l^k - \phi_j^k), \tag{9}$$

where dt for the CFL restriction is

$$dt = 1/\max_j \{\sum_{l \in N_j} w_{jl}\} \tag{10}$$

This process stops at some $k = K$th iteration chosen by the user.

3 Corners and Sharp Edges Detection

3.1 Overview

This section presents our corners and sharp edges detection method. The method is divided into two main steps, shown in Fig. 3. The first step is applying the level set method (LSM) to the data for segmentation in the form of implicit function. The data from the previous process is used to detect corners and sharp edges in the second step by convolving with designed masks, i.e. Sobel and Canny, respectively.

3.2 Corners Detection

In this step, marching cube is applied with each boundary voxel and Sobel-like mask is convolved in turn. Firstly, boundary voxels need to be filtered by processing a marching cube [15]. There are 15 configurations of a marching cube for polygonization, shown in Fig. 1 (a). Only two configurations are considered to filter the boundary voxel, which is possibly a corner voxel - or candidate corner voxel. That is, cases 1 and 5 are the cubes containing the corner candidate, shown in Fig. 1 (b). Instead of searching corners from the

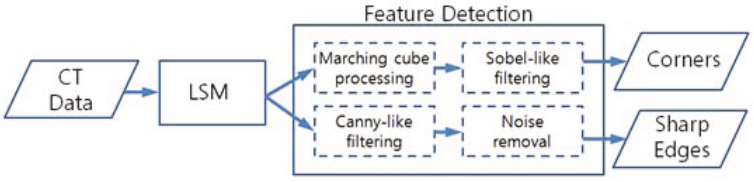


Fig. 3 Overview of Corners and Sharp Edges Detection

whole voxels, therefore, we only search corners from a set of corner candidates after processing a marching cube.

To detect a corner from the candidates, we use the method of 3D mask convolution. We invent 3D masks in the pattern of Sobel, shown in Fig. 5. There are three masks Sx, Sy and Sz provided in different directions, i.e. xy, yz, and xz-planes. We convolve these masks with a set of the corner candidates to estimate gradients in those three directions. Corners are voxels having small changes in directions of gradients or having extreme gradient. Pseudo code of detecting a corner is provided below.

```
Procedure cornerDetect(CC,Sx,Sy,Sz,ts1,ts2)
{
      Gx=Convolution(CC,Sx);
      Gy=Convolution(CC,Sy);
      Gz=Convolution(CC,Sz);
      Dxy=Gx/Gy ;
      Dyz=Gy/Gz ;
      Dzx=Gz/Gx ;
      For (i,j,k) in CC
      If( (|Dxy(i,j,k)-Dyz(i,j,k)|<ts1 &&
          |Dyz(i,j,k)-Dzx(i,j,k)|<ts1 &&
           |Dzx(i,j,k)-Dxy(i,j,k)|<ts1) ||
           (|Gx(i,j,k)|+|Gy(i,j,k)|+|Gz(i,j,k)|<ts2) )
         (i,j,k) is corner voxel.
}
```

where CC is a set of the corner candidates, ts1 and ts2 are thresholds, Dxy, Dyz and Dzx are directions of gradients in 3 axis directions.

3.3 Sharp Edges Detection

Likewise, we invent 3D masks to detect a sharp edge voxel. Like existing 3D edge detectors [10], [11], [14], this is used to approximate gradients or Laplacians of 3D images by using mask convolution. In our case, a 3D edge detector is inspired from Canny edge detector [3]. We design high pass filters

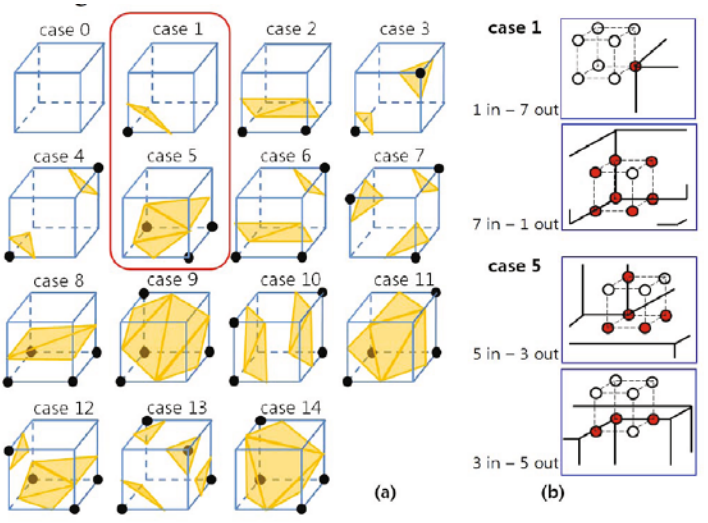


Fig. 4 (a)Configurations of a marching cube (b)Cases for corner candidate

to detect and extract sharp edge voxels by using convolution masks. Three masks, Cx, Cy and Cz are designed in three different directions, i.e. x-, y- and z-directions, and are shown in Fig. 6. They can detect convex sharp edge voxels. To detect concave sharp edge voxels, however, other three masks, Cx_i, Cy_i and Cz_i are also designed. The masks of Cy and Cy_i are shown Fig. 7. Pseudo code of detecting a sharp edge voxel is provided below.

```
Procedure edgeDetect(B,Cx,Cy,Cz,Cx_i,Cy_i,Cz_i,t1,t2)
{
        Gx=Convolution(B,Cx);
        Gy=Convolution(B,Cy);
        Gz=Convolution(B,Cz);
        Gx_i=Convolution(B,Cx_i);
        Gy_i=Convolution(B,Cy_i);
        Gz_i=Convolution(B,Cz_i);
        For (i,j,k) in B
         If ( Gx(i,j,k)<t1   || Gy(i,j,k)<t1   ||
              Gz(i,j,k)<t1   || Gx_i(i,j,k)>t2 ||
              Gy_i(i,j,k)>t2 || Gz_i(i,j,k)>t2  )
            (i,j,k) is a sharp edge voxel.
}
```

where B is a set of the boundary voxel, t1 and t2 are thresholds.

As well, this step includes a noise removal module. Because noise voxels and sharp edge voxels involve extreme gradients, we need to distinguish them. Susan method[13] is applied to solve this. This method is based on a circular

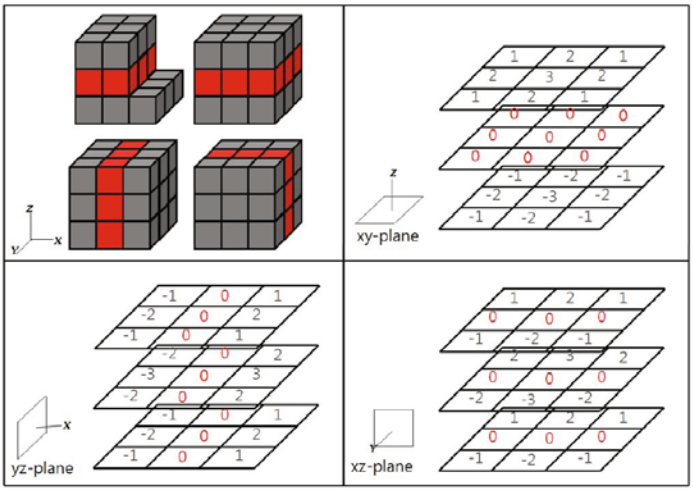


Fig. 5 Sobel-like pattern of convolution mask for detecting a corner.

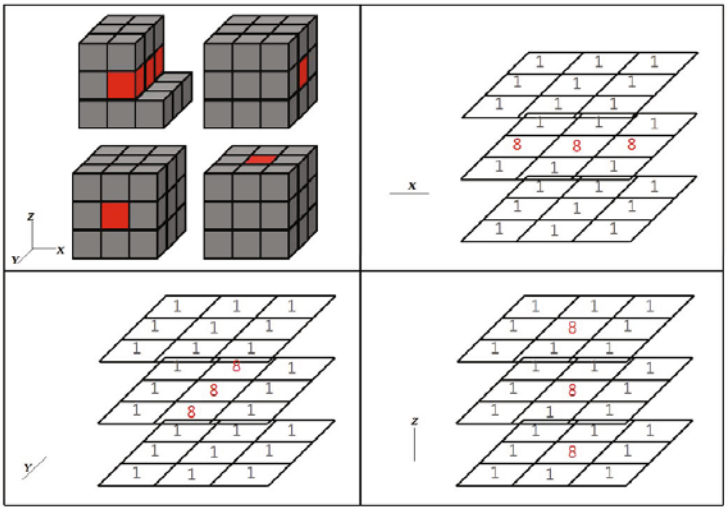


Fig. 6 Canny-like pattern of convolution mask for detecting a sharp edge.

window in which the central voxel, so-called nucleus, is the analyzed voxel. The operator responsibility is the ratio of the Susan area over the total area of the circular window. This ratio can be classified into (i) *salient* if it is less than 0.5, (ii) *flat* if it is approximately 0.5, and (iii) *concave*, otherwise. In this way, a salient is considered to a sharp edge voxel and a flat corresponds

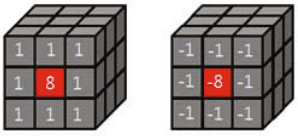


Fig. 7 Cy mask for convex sharp edge(left), Cy_i mask for concave sharp edge(right).

to noise. In order to compute the ratio, we use a cube window to determine total volume and interior volume, as shown in Fig. 8.

4 Experimental Results

We implement the algorithms mentioned in Section 3. A scanned data with resolution $50 \times 50 \times 50$ is tested and the results are shown in Fig. 9 with different thresholds setting. Red dots represent corner voxels, black dots sharp voxels and green dots boundary voxels of the shape. We can check the better result for corner detection in Fig. 9 (b) than Fig. 9 (a).

Fig. 10 shows the results of the scanned data with resolution of $300 \times 300 \times 250$. Fig. 10 (a) shows the visualization of the segmentation result from CT data. It is very noisy. After denoising using nonlocal means filter, we can get the result shown in Fig. 10 (b). This seems even better because lots of noise is reduced. But there are still some noises in surface. Fig. 10 (c) shows features extracted from (b) without SUSAN method. In this case, we can see some curves on the surface but not edge. After applying SUSAN method to (c), the result in Fig. 10 (d) is obtained. The curves from noises have gone away.

The computational cost of mask convolution is $O(n^3N)$, where $n \times n \times n$ is the size of a mask, N is the number of input data. In our method, since we extracted the information about surfaces from the volume data in the segmentation process, we have only to focus on the neighborhoods of the

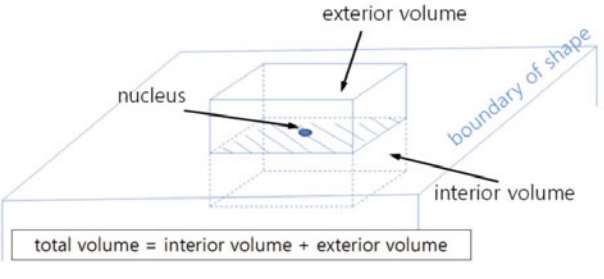


Fig. 8 A cube window for determining a SUSAN ratio.

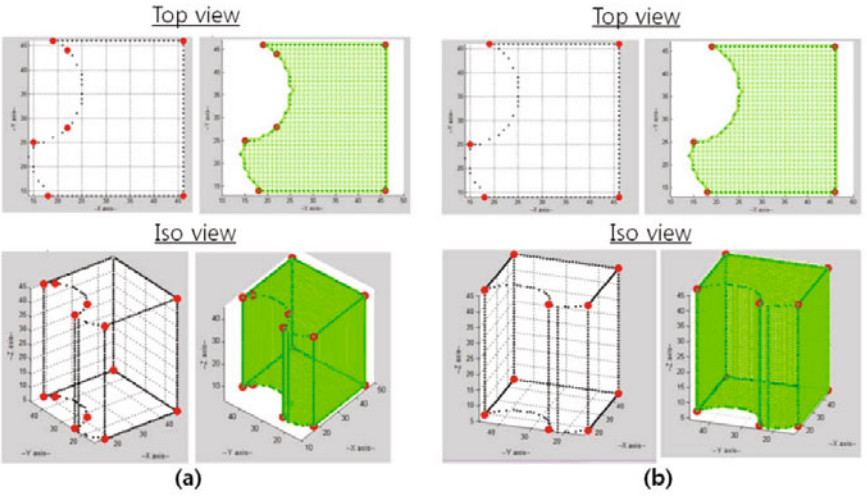


Fig. 9 (a)Thresholds : t1=15, t2=200, ts1=1.2, ts2=500 (b)Thresholds : t1=15, t2=100, ts1=0.7, ts2=350.

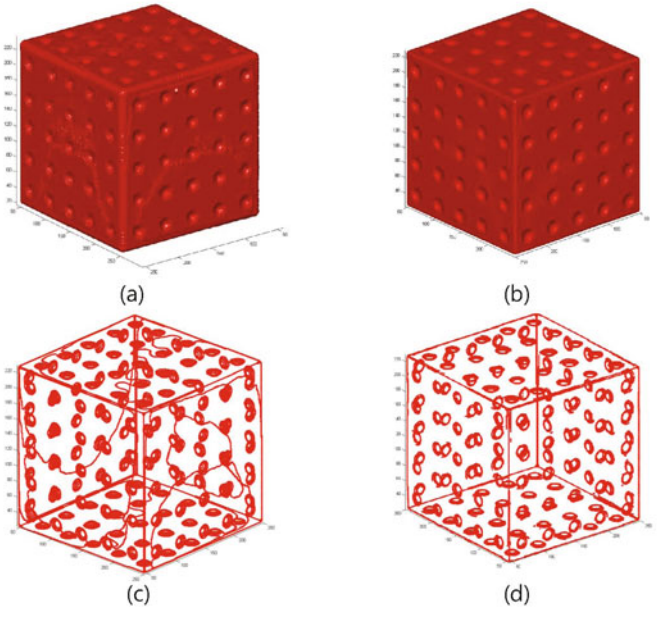


Fig. 10 Results of scanned data with resolution $300 \times 300 \times 250$.

surface. Thus the computational cost is $O(N) + O(n^3 N')$, where N' is the number of the boundary voxels, instead of $O(n^3 N)$. Because $N' \ll N$, the computational time with segmentation is faster.

5 Conclusion and Future Work

We propose a method of corners and sharp edges detection including two main steps. Firstly, the LSM method is applied to segment the boundary of the shape from 3D CT scanned data in form of implicit function. Secondly, nonlocal means filter is exploited to reduce noises. Finally, Sobel-like mask convolution with processing a marching cube and Canny-like mask convolution including a Susan noise removal module are done to detect corner voxels and sharp edge voxels, respectively. Computational cost is $O(N) + O(n^3 N)$, where N is the number of input data, $n \times n \times n$ is the size of a mask, and N' is the number of boundary voxels. The computational time is faster if we include the segmentation process.

For future work, we will develop a method for detecting other features such as ridges. Then we will develop how to construct a curve network from these resulting features and in the end reconstruct a B-spline model.

Acknowledgements. This work was supported by the Industrial Strategic technology development program, 10035474, Development of inspection platform technology based on 3-dimensional X-ray images funded by the Ministry of Knowledge Economy(MKE, Korea). M. Kang was supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program 2011 (211A5020021098501007).

References

1. Hubeli, A., Gross, M.: Multiresolution Feature Extraction for Unstructured Meshes. In: Proceedings of IEEE Visualization, pp. 287–294 (2001)
2. Song, B., Chan, T.: A Fast Algorithm for Level Set Based Optimization. CAM-UCLA 68 (2002)
3. Canny, J.: A Computational Approach to Edge Detection. TPAMI 8(6), 679–698 (1986)
4. Weber, C., Hahmann, S., Hagen, H.: Methods for Feature Detection in Point Clouds. In: Visualization of Large and Unstructured Data Sets - Applications in Geospatial Planning, Modeling and Engineering (IRTG 1131 Workshop) (2010)
5. Watanabe, K., Belyaev, A.G.: Detection of Salient Curvature Features on Polygonal Surfaces. Computer Graphics Forum, 385–392 (2001)
6. Demarsin, K., Vanderstraeten, D., Volodine, T., Roose, D.: Detection of Closed Sharp Edges in Point Clouds using Normal Estimation and Graph Theory. Computer-Aided Design 39(4), 276–283 (2007)
7. Hildebrand, K., Polthier, K., Wardetzky, M.: Smooth Feature Lines on Surface Meshes. In: Proceedings of 3rd Eurographics Symposium on Geometry Processing, pp. 85–90 (2005)

8. Pauly, M., Keiser, R., Gross, M.: Multi-scale Feature Extraction on Point Sampled Surfaces. Computer Graphics Forum 22, 281–289 (2003)
9. Monga, O., Deriche, R., Rocchisani, J.: 3D Edge Detection using Recursive Filtering: Application to Scanner images. CVGIP - Image Underst. 53(1), 76–87 (1991)
10. Morgenthaler, M., Rosenfeld, A.: Multidimensional Edge Detection by Hypersurface Fitting. IEEE Trans. Pattern Anal. Mach. Intell., 482–486 (1981)
11. Monga, O., Deriche, R., Malandain, G., Cocquerez, J.P.: Recursive Filtering and Edge Closing: Two Primary Tools for 3D Edge Detection. In: Proceedings of the First European Conference on Computer Vision (1990)
12. Gumhold, S., Wang, X., McLeod, R.: Feature Extraction from Point Clouds. In: Proceedings of 10th International Meshing Roundtable (2001)
13. Smith, S.M., Brady, J.M.: SUSAN - A New Approach to Low Level Image Processing. Int. J. Computer Vision 23(1), 45–78 (1997)
14. Zucker, S.W., Hummed, R.A.: A Three Dimensional Edge Operator. IEEE Transaction on Pattern Analysis and Machine Intelligence 3 (1981)
15. Lorensen, W.E., Cline, H.E.: Marching Cubes: A High Resolution 3D Surface Construction Algorithm. Computer Graphics 21(4) (1987)
16. Mumford, D., Shah, J.: Optimal Approximation by Piecewise Smooth Functions and Associated Variational Problems. Commun. Pure. Appl. Math 42, 577–685 (1989)
17. Chan, T., Vese, L.: Active Contours without Edges. IEEE Transactions on Image Processing 10(2), 266–277 (2001)
18. Osher, S., Fedkiw, R.: Level Set Method and Dynamic Implicit Surfaces. Springer (2003)
19. Dong, B., Ye, J., Osher, S., Dinov, I.D.: Level Set Based Nonlocal Surface Restoration. Multiscale Modeling and Simulation, 589–598 (2008)

Design of Lifecycle for Reliability of Open Source Software

Eun-Ser Lee and Joong-soo Kim

Abstract. This paper is intended to proposal lifecycle of open source software. There are many difficulty factors that cause the open source software problems during software interoperation. This paper evaluates the efficiency of lifecycle that detection of new risk items and remove ratio at the lifecycle of open source software.

Keywords: Open source software, Lifecycle, Reliable software.

1 Introduction

There is no doubt that the reliability of a computer program is an important element of its overall quality. If a program repeatedly and frequently fails to perform, it matters little whether other software quality factors are acceptable.

Software reliability, unlike many other quality factors, can be measured, directed, and estimated using historical and developmental data. Software reliability is defined in statistical terms as "the probability of failure free operation of a computer program in a specified environment for a specified time". To illustrate, program X is estimated to have a reliability of 0.96 over eight elapsed processing hours. In other words, if program X were to be executed 100 times and require eight hours of elapsed processing time (execution time), it is likely to operate correctly (without failure) 96 times out of 100[1][2][3].

Whenever software reliability is discussed, a pivotal question arises: What is meant by the term "failure"? In the context of any discussion of software quality and reliability, failure is nonconformance to software requirements. Yet, even

Eun-Ser Lee · Joong-soo Kim
Andong National University Computer Engineering 388 Songcheon-dong,
Andong-city, Gyeongsangbuk-do 760-749,
South Korea
e-mail: eslee@andong.ac.kr, kimjs@andong.ac.kr

R. Lee (Ed.): Software and Network Engineering, SCI 413, pp. 103–109.
springerlink.com

within this definition there are gradations. Failures can be only annoying or catastrophic. One failure can be corrected within seconds while another re-quires weeks or even months to correct. Complicating the issue even further, the correction of one failure may in fact result in the introduction of other errors that ultimately result in other failures.

This paper is proposal the lifecycle to reduce of risk items and guarantee the reliable software in the project progress. Also, this paper is intended to develop the relationship between defects and their causes to introduce. Also, this paper is intended to develop the relationship between process improvement and their causes to introduce.

2 Related Works

2.1 Defect Removal Efficiency

The DRE(Defect Removal Efficiency) of the defect detection stage is the ratio of the number of defects discovered at a stage against the total number of defects that appears when the stage is being processed. The more effective the DRE, the lower the possibility of undetected defects[4][8][9][10]. This shows that increasing the DRE(Defect Removal Efficiency) is a method to improve productivity.

DRE is calculated as follows.

DRE = E/(E+D)

E= Number of defect found at relevant S/W development step(e.g : Number of defect found at request analysis step)

D= Number of defect found at next S/W development step (e.g : Defect number that defect found at design step is responsible for defect of request analysis step)

Ideal value of DRE is 1 or 0(no error), and this displays that no defect on the project.

2.2 Quality Factors

The factors that affect software quality can be categorized in two broad groups: (1) factors that can be directly measured (e.g., usability or maintain-ability). In each case measurement must occur. We must compare the software (documents, programs, data) to some datum and arrive at an indication of quality.

McCall and his colleagues proposed a useful categorization of factors that affect software quality. These software quality factors, focus on three important aspects of a software product: its operational characteristics, its ability to undergo change, and its adaptability to new environments[5][6][7].

McCall provides the following descriptions:

1) Correctness. The extent to which a program satisfies its specification and fulfills the customer`s mission objectives.

2) Reliability. The extent to which a program can be expected to perform its intended function with required precision. It should be noted that other, more complete, definitions of reliability have been proposed.
3) Efficiency. The amount of computing resources and code required by a program to perform its function.
4) Integrity. The extent to which access to software or data by unauthorized persons can be controlled.
5) Usability. The effort required to learn operate prepare input, and interpret output of a program.
6) Maintainability. The effort required to locate and fix and error in a program. (This is a very limited definition.)
7) Flexibility. The effort required to modify an operational program.
8) Testability. The effort required to test a program to ensure that it performs its intended function.
9) Portability. The effort required to transfer the program from one hardware and/or software system environment to another.
10) Reusability. The extent to which a program [or parts of a program] can be reused in other applications related to the packaging and scope of the functions that the program performs.
11) Interoperability. The effort required to couple one system to another.

$$F_q = c_1 \times m_1 \times c_2 \times m_2 + \cdots + c_n \times m_n$$

It is difficult, and in some cases impossible, to develop direct measures of the above quality factors.

3 Theory and Case Study

This chapter will be proposed a level to identify capability of security factor that needed during actual systems. Therefore, the structure and contents to improve the items and analyze the result are presented in this chapter.

In this paper is using published paper data. We are already published other paper about the security lifecycle. Section 3.1 is published data. And, the content is like the following.

There are many methodologies for providing reliable software system and engineering. Therefore we need to manage system and methodology.

Lifecycle depicts the relationship between the customer's requirements and the engineer. There are many Lifecycle is used to provide a context for discussion and should not be construed as advocating a preference for one methodology (e.g. waterfall) over another (e.g. prototyping)[8][9][10].

In this paper is proposal the lifecycle that reduce the error and interoperate between legacy software and new software.

3.1 Definition of Requirement of Software Development Process

We must analyze the customer's requirements for the lifecycle of open source software. Therefore provide the requirements of life cycle that interoperation of open source software.

Table 1 Description of the protection profile

	Requirement	Constraint
Commonality	- Provide reliability of interoperation among functions	- Analysis and specification of attribute
Variability	- Influence of other function and changeable of attribute and method	- Maintain data Integrity - Design of structure for variability
System	- Provide hardware and software of interoperation methodology	- Specification of hardware and software performance
Function	- Provide of interface to functions and components	- Standard of interface type
Interoperation	- Provide reliability of interoperation among functions	- Considering of low coupling and high cohesion

1) Commonality – The commonality is view point of dependent function for the open source software. Therefore, it is Provide reliability of interoperation among functions.
2) Variability - The variability is view point of independent function for the open source software. Therefore, it is influenced the function and changeable of attribute and method
3) System - The system is view point of Hardware and software for the open source software. Therefore, it is Provide hardware and software of interoperation methodology
4) Function - The function is view point of using for each of the open source software. Therefore, it is Provide of interface to functions and components.
5) Interoperation - The interoperation is view point of reliable performance for each of the open source software. Therefore, it is Provide reliability of interoperation among functions.

3.2 Lifecycle of Open Source Software

In this paragraph is proposal lifecycle of open source software during development system. Each of the factors is use of the repeatable milestone for the progress management. Factor of lifecycle is like the following.

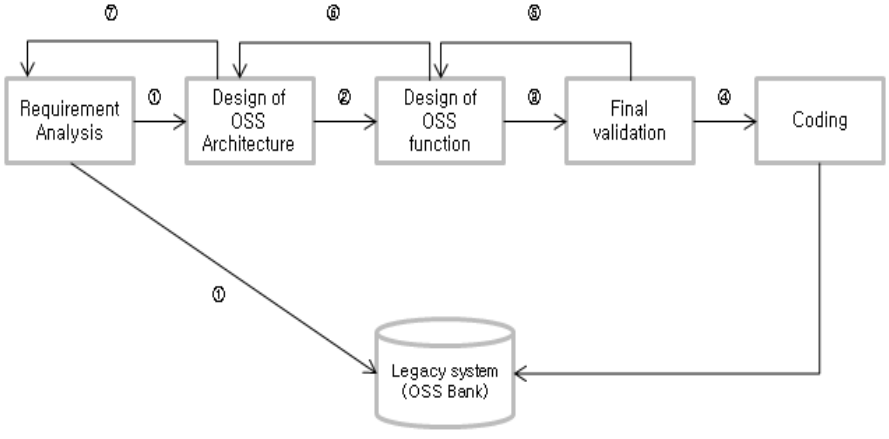


Fig. 1 Lifecycle of open source software

Fig. 1 is show the representation of lifecycle from requirement to coding for the open source software. Each of stages is like the next step.

① Search for the correct function in the OSS (open source software) bank based on the requirements.
② If function is not matching, develop the OSS (open source software)
③ Evaluation the suitability of structure that OSS (open source software) of reflect the requirement
④ Design of the structure that OSS (open source software) of reflect the requirement for the proper performance
⑤ If, in case of requirement and function is not satisfied, you must be feedback in the final validation
⑥ If, in case of requirement is not satisfied and have a error of architecture, you must be feedback
⑦ If, in case of requirement is not satisfied and have a error of analysis, you must be feedback

3.3 Evaluation on Lifecycle of OSS (Open Source Software)

In order to lifecycle of OSS (Open Source Software) proposal to improves quality of product and heighten productivity. Therefore, when we applied lifecycle in the development of open source software, we wish to apply defect removal efficiency to measure ability of defect control activity.

With apply lifecycle, defect removal efficiency analysis [15] investigated defect number found at relevant S/W development step and defect number found at next time step in terms of request analysis, design and coding stage. We show you to compute the defect removal efficiency is as follows

Table 2 Tabel of defect removal efficiency

	Number(%) of defect found at relevant S/W development step (E)	Number(%) of defect found at next S/W development step (D)
Requirement Analysis	60	12
Design of OSS Architecture	30	6
Design of OSS function	14	3
Final validation	7	1
Coding	3	1
Total	114	23

Table 2 is a table to show up defect number on the each step by inspection with defect trigger application. inspect software development step defect number after Defect Trigger application. We get DRE at each software development step by table 4, it is as following.

0.833 = 60 / (60+12) (Requirement Analysis)
0.833 = 30 / (30+6) (Design of OSS Architecture)
0.824 = 14 / (14+3) (Design of OSS function)
0.875 = 7 / (7+1) (Final validation)
0.75 = 3 / (3+1) (Coding)

Therefore, because DRE is approximated to 1, when we remove defect by lifecycle, DRE was analyzed good efficiency.

4 Conclusion

In this paper we are proposed the lifecycle of open source software applicable. Also, in this paper provide the criteria for judgment in the lifecycle of open source software. This lifecycle is able to the various area. Therefore, various systems and types is require the architecture for the flexibility and usability of the support.

For the future studies will be provide the development tool for the applicable to extract of security requirement and evaluate the architecture of the personal security. And we are supplements architecture after applying of the real world. Also, we are analysis of the architecture efficiency against security of the defense and flexibility of rate. Therefore, we are put to use the reliable method of the evaluation.

References

1. Garfinkel, S., Spafford, G.: Web security, Privacy and commerce. O'Reilly & Associates (2002)
2. ISO. ISO/IEC 15408-2: Information technology - Security techniques - Evaluation criteria for IT security - Part 2: Security functional requirements (1999)

3. ISO. ISO/IEC 15408-3: Information technology - Security techniques - Evaluation criteria for IT security - Part 3: Security assurance requirements (1999)
4. ISO/IEC Guide 65—General Requirements for Bodies Operating Product Certification Systems (1996)
5. Pressman, R.S.: A practice's approach, 6th edn. (2005)
6. Lee, E.-S., Lee, K.W., Kim, T.-H., Jung, I.-H.: Introduction and Evaluation of Development System Security Process of ISO/IEC TR 15504. In: Laganá, A., Gavrilova, M.L., Kumar, V., Mun, Y., Tan, C.J.K., Gervasi, O. (eds.) ICCSA 2004. LNCS, vol. 3043, pp. 451–460. Springer, Heidelberg (2004)
7. Software Process Improvement Forum, KASPA SPI-7 (December 2002)
8. Dunn, R.H.: Software defect removal. McGraw-hill (1984)
9. Fenton, N., Ohlsson, N.: Quantitative analysis of faults and failures in a complex software system. IEEE Trans. Software Eng. 26, 797–814 (2000)
10. Lee, E., Lee, K.W., Lee, K.: Design Defect Trigger for Software Process Improvement. In: Ramamoorthy, C.V., Lee, R., Lee, K.W. (eds.) SERA 2003. LNCS, vol. 3026, pp. 185–208. Springer, Heidelberg (2004)

Evaluation of Risk Items for Open Source Software

Eun-Ser Lee and Haeng-Kon Kim

Abstract. This paper is intended to evaluate the risk items of lifecycle for the open source software. There are many difficulty factors that cause the open source software problems during software interoperation. Also, using defect cause, we understand associated relation between defects and design defect trigger. So when we archive correspond project, we can forecast defect and prepare to solve defect by using defect trigger. This paper evaluates the degree of risk of lifecycle that detection of new risk items and remove ratio at the lifecycle of open source software.

Keywords: Defect, Defect Trigger, Risk, Open source software, Lifecycle.

1 Introduction

When risk is considered in the context of software engineering, Charette's three conceptual underpinnings are always in evidence. The future is our concern — what risks might cause the software project to go awry? Change is our concern — how will changes in customer requirements, development technologies, target computers, and all other entities connected to the project affect timeliness and overall success? Last, we must grapple with choices — what methods and tools should we use, how many people should be involved, how much emphasis on quality is "enough?"

Reactive risk strategies have been laughingly called the "Indiana Jones school of risk management". In the movies that carried his name, Indiana Jones, when

Eun-Ser Lee
Andong National University Computer Engineering 388 Songcheon-dong,
Andong-city, Gyeongsangbuk-do 760-749, South Korea
e-mail: eslee@andong.ac.kr

Haeng-Kon Kim
Dept. of Computer Information & Communication Engineering,
Catholic University of Deagu, Korea
e-mail: hangkon@cu.ac.kr

R. Lee (Ed.): Software and Network Engineering, SCI 413, pp. 111–117.
springerlink.com

faced with overwhelming difficulty, would invariably say, “Don’t worry, I’ll think of something!” Never worrying about problems until they happened, Indy would react in some heroic way[5][8][9].

A considerably more intelligent strategy for risk management is to be proactive. A proactive strategy begins long before technical work is initiated. Potential risks are identified, their probability and impact are assessed, and they are prioritized by importance. Then, the software team establishes a plan for managing risk. The primary objective is to avoid risk, but because not all risks can be avoided, the team works to develop a contingency plan that will enable it to respond in a controlled and effective manner. Throughout the remainder of this chapter, we discuss a proactive strategy for risk management[1][2].

This paper is estimate of the lifecycle to reduce of risk items and guarantee the reliable software in the open source software. Also, this paper is intended to develop the relationship between defects and their causes to introduce.

2 Related Works

2.1 Defect Trigger

Traditionally, defects represent the undesirable aspects of a software's quality. Root Cause Analysis (RCA)and Statistical Growth Modeling (e.g. S-curves) have played useful roles in the analysis of software defects. Effective RCA, while yielding exhaustive details on each defect, takes substantial investment of resources for completion and points to too many actions as a result. Growth modeling, on the other hand, provides an easy way to monitor trends, but is not capable of suggesting corrective actions due to the inadequate capture of the semantics behind the defects[25]. Trigger is a scheme to capture the semantics of each software defect quickly. It is the definition and capture of defect attributes that make mathematical analysis and modeling possible. Analysis of trigger data provides a valuable diagnostics method for evaluating the various phases of the software life cycle and the maturity of the product[3].

2.2 Risk Identification

Risk identification is a systematic attempt specify threats to the project plan(estimates, schedule, resource loading, etc.). By identifying known and predictable risks, the project manager takes a first step toward avoiding them when possible and controlling them when necessary. Product-specific risks can only be identified by those with a clear understanding of the technology, the people, and the environment that is specific to the project at hand. To identify examined and an answer to the following question is developed[7].

Both generic and product-specific risks should be identified systematically. Tom Gilb drives this point home when he states: “If you don’t actively attack the risks, they will actively attack you.”

One method for identifying risks is to create a risk item checklist. The checklist can be used for risk identification and focuses on subset of known and predictable risks[4][5].

3 Theory and Case Study

This chapter will analyze and measure the identified defect causes of lifecycle for each of development stages. To this end, the hierarchical structure for decision-making has been schematized, and the weight of the defect item in each stage has been graded into six dimensions. The graded items were analyzed based on a comparison metrics [6][10], and the geometric means of the defect items were calculated in order to identify its relation with the causes.

3.1 Elicitation of Risk Items

This chapter will be proposal to elicitation of risk items. A risk items detection trigger was made to identify risk in the development project.

Risk items were divided as below by classifying defects that exist in each lifecycle into open source software. The structure of risk items for each lifecycle is as follows.

To detect risk items, five detailed categories were made for the risk items at the requirement analysis stage. The total structure is as follows.

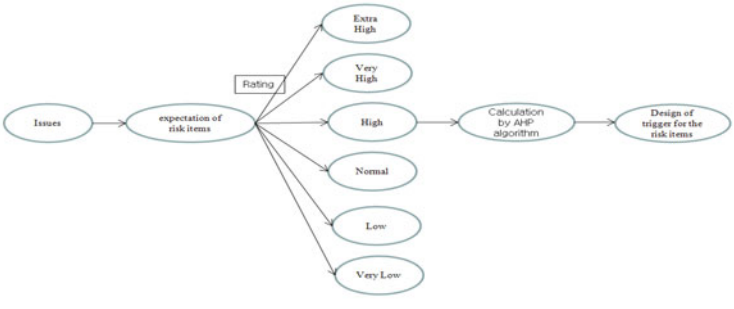


Fig. 1 Process of structure for the elicitation of risk items

The decision-making structure and graded items in each stage are as below.

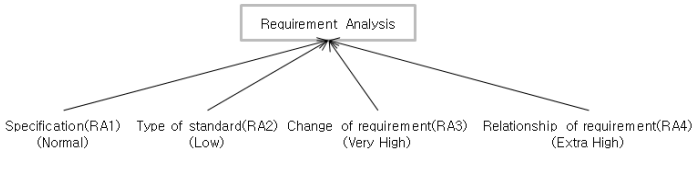


Fig. 2 Cause rating scale of risk in requirement analysis stage

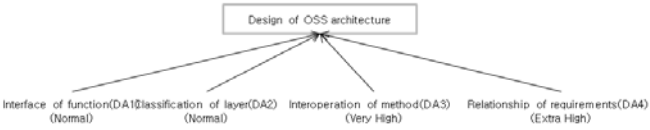


Fig. 3 Cause rating scale of risk in Design of open source software stage

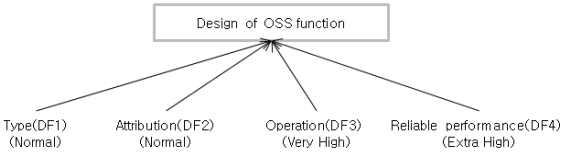


Fig. 4 Cause rating scale of risk in Design of open source function stage

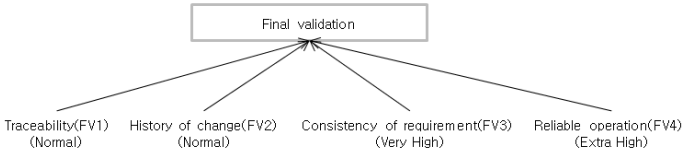


Fig. 5 Cause rating scale of risk in final validation stage

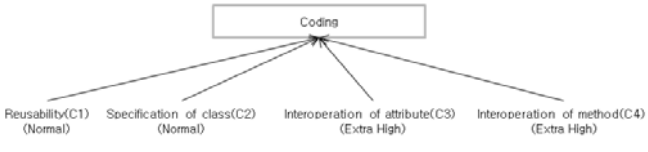


Fig. 6 Cause rating scale of risk in coding stage

The risk cause rating scale structure is divided into five stages (Requirement analysis, Design of OSS architecture, Design of OSS function, Final validation, Coding).

The rating scale is divided into six dimensions: Very Low(VL), Low(L), Normal(N), High(H), Very High (VH), Extra High (EH). This means that, among the six dimensions, priority increases as it nears EH, and the priority decreases when as it approaches VL.

3.2 Risk Analysis

A matrix was used to compare the defect causes. The matrix shows the relation by listing each importance from 1 to 9 [6][10]. The measurement standard table of the Comparison Matrix is shown in table 1.

The importance among the items of each stage were quantified based on table 1. Figure 1 to 6 were analyzed by table 1. This time, the differences of the item grade of each stage were compared. For example, if an item is EH and the other item is N, the importance between the two items is triple the difference. This is due to two grade differences. The same relation applies to table 1.

Table 1 Cause ratio production table of Grade vs importance difference

Grade difference between item	Grade vs importance difference ratio
1	Double
2	Triple
3	Quadruple
4	Quintuple
5	Sextuple

By table 2, it is possible to produce a comparison matrix of figure 1. The contents are the same with table 2.

Table 2 Comparative matrix between items of requirements analysis stage

	RA1	RA2	RA3	RA4
RA1	1	1/2	3	4
RA2	2	1	4	5
RA3	1/3	1/4	1	2
R 4	1/4	1/5	1/2	1

According to the analysis, between RA1 and RA2, RA1's defect item is double that of RA2's. This is because a11 is Normal, and a12 is Very Low. Therefore, the difference of grade between items is 1 by table 1. Therefore, the difference between the grade and importance is double. This implies that the probability that RA1 will cause a defect is twice as high as RA2.

RA2's risk degree is higher than other items. (2 x 1 x 4 x 5) / 4 x 100 = 800

Table 3 Comparative matrix between items of design of open source software architecture stage

	DA1	DA2	DA3	D4
DA1	1	1	1/3	1/4
DA2	1	1	1/3	1/4
DA3	1/3	1/3	1	1/2
DA4	1/4	1/4	1/2	1

DA3's risk degree is higher than other items. (1/3 x 1/3 x 1 x 1/2) / 4 x 100 = 1.36

Table 4 Comparative matrix between items of design of open source software function stage

	DF1	DF2	DF3	DF4
DF1	1	1	1/3	1/4
DF2	1	1	1/3	1/4
DF3	1/3	1/3	1	1/2
DF4	1/4	1/4	1/2	1

DF3's risk degree is higher than other items. (1/3 x 1/3 x 1 x 1/2) / 4 x 100 = 1.36

Table 5 Comparative matrix between items of final validation stage

	FV1	FV2	FV3	FV4
FV1	1	1	1/3	1/4
FV2	1	1	1/3	1/4
FV3	1/3	1/3	1	1/2
FV4	1/4	1/4	1/2	1

FV3's risk degree is higher than other items. (1/3 x 1/3 x 1 x 1/2) / 4 x 100 = 1.36

Table 6 Comparative matrix between items of coding stage

	C1	C2	C3	C4
C1	1	1	1/3	1/4
C2	1	1	1/3	1/4
C3	1/3	1/3	1	1/2
C4	1/4	1/4	1/2	1

C3's risk degree is higher than other items. (1/3 x 1/3 x 1 x 1/2) / 4 x 100 = 1.36

4 Conclusion

In this paper is presented the designing and analysis of a risk items. Also, in this paper evaluate the risk items for the lifecycle of open source software. This lifecycle is able to the various field.

For the future studies will be provide the smart application for the open source software based android and IOS. Therefore, we are put to use the reliable method of the evaluation.

References

1. Garfinkel, S., Spafford, G.: Web security, Privacy and commerce. O'Reilly & Associates (2002)
2. Kratschmer, T.: Improving Education of Software Engineers Through Use of Defect Analysis. Submitted to IEEE Software Magazine (September/October 2002)

3. Fenton, N., Neil, M.: Software Metrics: Successes, Failures, and New Directions. J. Systems and Software 47, 149–157 (1999)
4. Gaffney, J.: Some Models for Software Defect Analysis. Lockheed Martin (November 1996)
5. Pressman, R.S.: A practice's approach, 6th edn. (2005)
6. Lee, E.-s., Lee, K.W., Kim, T.-h., Jung, I.-H.: Introduction and Evaluation of Development System Security Process of ISO/IEC TR 15504. In: Laganá, A., Gavrilova, M.L., Kumar, V., Mun, Y., Tan, C.J.K., Gervasi, O. (eds.) ICCSA 2004. LNCS, vol. 3043, pp. 451–460. Springer, Heidelberg (2004)
7. Software Process Improvement Forum, KASPA SPI-7 (December 2002)
8. Dunn, R.H.: Software defect removal. McGraw-Hill (1984)
9. Fenton, N., Ohlsson, N.: Quantitative analysis of faults and failures in a complex software system. IEEE Trans. Software Eng. 26, 797–814 (2000)
10. Lee, E., Lee, K.W., Lee, K.: Design Defect Trigger for Software Process Improvement. In: Ramamoorthy, C.V., Lee, R., Lee, K.W. (eds.) SERA 2003. LNCS, vol. 3026, pp. 185–208. Springer, Heidelberg (2004)

Cloud Computing for Business

Jan Seruga and Ha Jin Hwang

Abstract. Advanced information technologies and the Internet have resulted in the emergence of the phenomenon of processing in the clouds (cloud computing - CC). A general definition of processing in the clouds is "access to a resource on the Internet outside the company firewall."

This work is an introduction to a rapid development inweb application functionality going beyond the traditional concept of Web 2.0 applications. The new applications of cloud computing bring mobility and are delivered to all possible devices designed to interface with the user, ranging from PCs to smart phones. The aim of this review is to investigate the potential of these new solutions to combine the advantages of desktop applications (speed, ergonomics, user friendliness, access to local resources) and web applications (mobility, accessibility, scalability).

Keywords: cloud computing, processing.

1 Working in the Internet Environment

Cloud computing as a delivery model for IT services is defined by the National Institute of Standards and Technology (NIST) as “a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”. [5]

McCarthy predicted in the 1960s that the processing of data will be organized in a network pattern similar topublic services such as electricity and water. Another visionary, Ian Foster predicted in the 90s that computing power would be

Jan Seruga
Australian Catholic University
e-mail: jan.seruga@acu.edu.au

Ha Jin Hwang
Kazakhstan Institute of Management, Economics, and Strategic Research(KIMEP)
e-mail: hjhwang@kimep.kz

R. Lee (Ed.): Software and Network Engineering, SCI 413, pp. 119–132.
springerlink.com

provided to end users (consumers) on request. These two visions have found their fulfilment in a world of technological solutions related to CC. [2]

Company employees need to be increasingly mobile and have access to corporate resources anytime and anywhere. Until recently the only way to safely use the company's internal resources was the through VPN technology. However VPN devicesare expensive, add burden to network traffic and require purchase of a license for each user. [3]

Specific factors that initiated changes in the perception of the Internet as a medium for the processing of corporate information are as follows:

- Broadband Internet
- Social networking
- Increasing number ofdevices with Internet access,
- The expectations of users
- Economies of scale[2]

While cloud vendors focus on TCO (Total Cost of Ownership) as the primary benefit of cloud solutions, customers are most interested in business agility. A recent survey of 500 IT decision-makers by SandHill found that ~50% of respondents cited business agility as their primary reason for adopting cloud applications. Similarly, an InformationWeek survey found that 65%+ of respondents indicated that responding faster to the business is an important driver for cloud computing. Hence being able to respond more quickly to changing business requirements is seen as a critical advantage of the cloud model.[9]

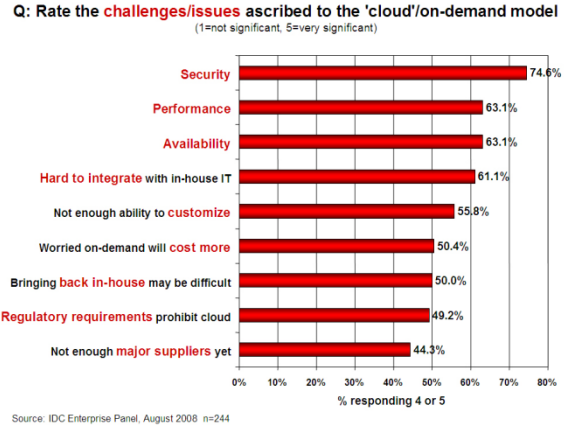


Fig. 1 Cloud computing and challenges.

NIST specify five characteristics of cloud computing as follows:

a. On-demand self-service involves customers using a web site or similar control panel interface to provision computing resources such as additional computers, network bandwidth or user email accounts, without requiring human interaction between customers and the vendor.

b. Broad network access enables customers to access computing resources over networks such as the Internet from a broad range of computing devices such as laptops and smartphones.

c. Resource pooling involves vendors using shared computing resources to provide cloud services to multiple customers.

d. Rapid elasticity enables the fast and automatic increase and decrease to the amount of available computer processing, storage and network bandwidth as required by customer demand.

e. Pay-per-use measured service involves customers only paying for the computing resources that they actually use, and being able to monitor their usage. This is analogous to household use of utilities such as electricity. [10]

By virtualizing their resourcesa cloud can provide a runtime environment for operating systems and applications in multi-tenant mode.[4] The introduction of clouds means transforming the IT departments into service departments and/or providing the ability to secure data storage and backup. Gartner has analysed the current state of information technology on the marketwith some interesting conclusions that 8 out of 10 U.S. dollars in an IT budget for expenses related to the technology goes to the maintenance of existing systems rather than on innovation.[6]

Accessinga service in the clouds requires registration. A resource is available if it presents a certificate of identity which allowsidentifying the person or process, and on this basis, authorizes the use of it. This process is called authenticationand consists only of identifying "who you are"- not to be confused with authorization. A token, certificate, fingerprint are attributes that can uniquely identify the person. A new trend in logging is to use an existing account in another site, e.g. log in via Facebook. The user can login to the service quickly and easily withthe use a single identity certificate to a variety of important services.

2 Processing in the Clouds

A further reason for processing in the clouds besides the economic benefits is the underutilisation of increasingly powerful company data servers which are needed only in certain periods, eg processing the monthly billing. The IT department buys servers in order to cope with temporary, but repeated surges in demand for computing power of other departments. Research has shown that on average only 10% of the computing capacity of data servers is utilised. [12]. Cloud computing has become increasingly popular due to the ability to provide a flexible dynamic IT infrastructure.[10]

What does processing in the cloud do for the end user?

- An individual usercan store their private resources in a secure site with minimal resources and from any place on earth. Such services as email, calendar, bookmarks, browsers, documents and notes can be easily shared if required.

- Group business users will have access to applications supporting their business processes like Customer Relations Management, Human Resources, and Accounting.
- Improved responsiveness to customer needs and the functioning of the IT department.

There are four cloud deployment models[1] as shown in figure 2:

a. Public cloud involves an organisation using a vendor's cloud infrastructure which is shared via the Internet with many other organisations and other members of the public. This model has maximum potential cost efficiencies due to economies of scale. However, this model has a variety of inherent security risks that need to be considered.

b. Private cloud involves an organisation's exclusive use of cloud infrastructure and services located at the organisation's premises or offsite, and managed by the organisation or a vendor. Compared to the public cloud model, the private cloud model has reduced potential cost efficiencies. If the private cloud is properly implemented and operated, it has reduced potential security concerns. A well architected private cloud properly managed by a vendor provides many of the benefits of a public cloud, but with increased control over security. A managed private cloud may enable enterprise customers to more easily negotiate suitable contracts with the vendor, instead of being forced to accept the generic contracts designed for the consumer mass market that are offered by some public cloud vendors.

c. Community cloud involves a private cloud that is shared by several organisations with similar security requirements and a need to store or process data of similar sensitivity. This model attempts to obtain most of the security benefits of a private cloud, and most of the economic benefits of a public cloud. An example community cloud is the sharing of a private cloud by several agencies of the same government.

d. Hybrid cloud involves a combination of cloud models. An example is using commodity resources from a public cloud such as web servers to display non-sensitive data, which interacts with sensitive data stored or processed in a private cloud.

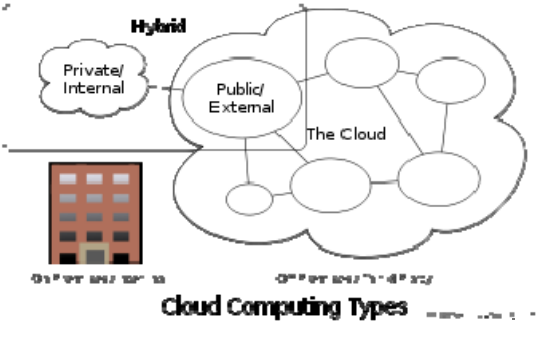


Fig. 2 Types of Clouds[13]

Cloud computing has the potential to help agencies leverage modern technologies such as computer virtualisation and worldwide Internet connectivity. Some of the key business drivers are:

a. Pursuing new business opportunities, such as trialling new ideas to reach and interact with customers over the Internet;

b. Reducing upfront costs of capital expenditure of computer equipment and related expenses such as a physical data centre and support staff, while reducing the associated financial risk to the agency by replacing upfront costs with reasonably predictable operational expenditure, and only paying for the amount of computing processing and data storage that is actually used;

c. Potentially reducing ongoing costs due to the use of infrastructure and technical specialists that are typically shared among many customers to achieve economies of scale, however the cost of applying controls to help address security risks especially associated with shared infrastructure may reduce the potential cost savings of some types of cloud computing;

d. Potentially improving business continuity and the availability of computing infrastructure if users have guaranteed available network connectivity, where the infrastructure can rapidly and flexibly scale to meet peaks and troughs in usage demand, and with the computing infrastructure typically located in multiple physical locations for improved disaster recovery; and,

e. Potentially reducing carbon footprint due to the more efficient use of computer hardware requiring less electricity and less air conditioning.

There are three cloud service models. A non-exhaustive list of example vendor services is provided to help the reader understand the cloud service models. Inclusion of an example vendor service does not imply DSD's support of the service.

a. Infrastructure as a Service (IaaS) involves the vendor providing physical computer hardware including CPU processing, memory, data storage and network connectivity. The vendor may share their hardware among multiple customers referred to as "multiple tenants" using virtualisation software. IaaS enables customers to run operating systems and software applications of their choice. Typically the vendor controls and maintains the physical computer hardware. Typically the customer controls and maintains the operating systems and software applications.

Example IaaS vendor services include Amazon Elastic Compute Cloud (EC2), GoGrid and Rackspace Cloud.

b. Platform as a Service (PaaS) involves the vendor providing Infrastructure as a Service plus operating systems and server applications such as web servers. PaaS

enables customers to use the vendor's cloud infrastructure to deploy web applications and other software developed by the customer using programming languages supported by the vendor. Typically the vendor controls and maintains the physical computer hardware, operating systems and server applications. Typically the customer only controls and maintains the software applications developed by the customer. Example PaaS vendor services include Google App Engine, Force.com, Amazon Web Services Elastic Beanstalk, and the Microsoft Windows Azure platform.

c. Software as a Service (SaaS) involves the vendor using their cloud infrastructure and cloud platforms to provide customers with software applications. Example applications include email and an environment for users to collaboratively develop and share files such as documents and spreadsheets. These end user applications are typically accessed by users via a web browser, eliminating the need for the user to install or maintain additional software. Typically the vendor controls and maintains the physical computer hardware, operating systems and software applications. [2]

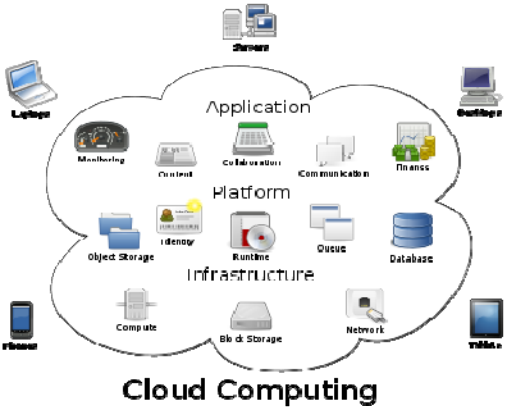


Fig. 3 Cloud computing logical diagram. [20]

Typically the customer only controls and maintains limited application configuration settings specific to users such as creating email address distribution lists. Example SaaS vendor services include Salesforce.com Customer Relationship Management (CRM), Google Docs and Google Gmail. Microsoft Office 365 (formerly called Business Productivity Online Suite) consists of Microsoft Office Web Apps, Microsoft Exchange Online, Microsoft SharePoint Online, Microsoft Dynamics CRM Online and Microsoft Lync. [15]

3 Introducing Cloud Computing to the Enterprise

The introduction of CC to the enterprise is difficult because of the need to calculate the ROI (Return of Investment) and to demonstrate the viability of processing in the clouds from the perspective of business needs. The following are key advantages of CC for enterprise [2]:

1. The ability to create the illusion of access to an infinite computing power and capacity, regardless of the growth from one user to 100 or from one to 1000 –great scalability
2. Abstraction from implementation details and infrastructure,
3. The principle of pay for consumption (such as a water meter) in relation to IT services with little or no initial charges
4. Using the service by connecting the device to a resource
5. The service is on demand, scalable upwards and downwards, and available immediately i.e. there is no need to book in advance or plan the time of usage
6. Access to applications and information with any Access Point

These are not features unique to this technology. What really distances it from others is the range of changes, the size of cost reductions and the impact on processing performance. The efficiency and effectiveness can be improved in the order of 5 to 10 times.This can be traced back to the famous graph of the curve " Capacity-Utilization ", as shown in figure 4, first presented by Amazon.[14] Curve CU (Capacity-Utilization) has become an icon, a pillar of justifying the introduction and use of the CC, in which the current needs are met by the immediate availability of on-demand business services.

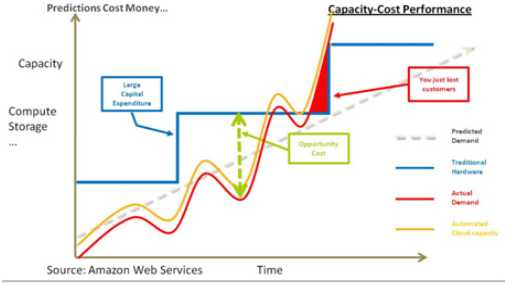


Fig. 4 The Capacity versus Utilization Curve

There is a trend among providers of CC - instead of offering virtualization in a cloud of commonly used applications (CRM, ERP) it is proposed to transfer to the clouds concrete, low-level operations related to IT such as backup service, mail and file archive.

4 Cloud Computing and Corporate Business Model

It is likely that more and more software will be provided as a service via cloud rather than a traditional software purchase and licensing.IDC survey of enterprises in Australia found that 20.6 per cent of the respondents are using Cloud computing, while 32.4 per cent are planning to deploy Cloud services in the next six to 12 months. A further 41.2 per cent of companies are planning to implement Cloud services by 2013.[15]

The IT department can take advantage of CC because it provides scalability and immediate availability (instant provisioning). The IT department has an important task - to ensure continuity of the equipment, networks and applications (disaster recovery service). In times of crisis it is often this task that falls victim to cost savings. Instead of maintaining expensive backup centres the virtualization infrastructure can be used to secure continuity in the event of failure. Service providers like Microsoft and Google offer an option to transfer applications in the cloud environment without the use of local servers. Microsoft offers Azure which is supported by the development environment of Visual Studio 2010 with multiple languages like C #, VB.NET, Python, Ruby and PHP. It offers App Engine which supports Java and Python.

Cloud Service providers[16]:

Amazon

Specializes in IaaS. It offers through Amazon Web Services (**AWS**): Amazon **Elastic Compute Cloud** (EC2), Amazon **SimpleDB**, Amazon **CloudFront**, Amazon **SQS**. It claims to have 82 billion objects stored in Amzon**S3** (Simple Storage Service)

Google

Specializes in **PaaS**&**SaaS**. As **SaaS** it offers **Google Apps:** a web-based communication, collaboration & security apps which includes, Gmail, Google Calendar, Google Talk, Google Docs & Google Sites. As PaaS it offers Google **App Engine:** a platform for developing and hosting web applications in Google-managed data centers. Currently, the supported programming languages are *Python* and *Java* (by extension other JVM languages are also supported).

VMware

VMware offers **vCloud**: this canrun, secure and manage applications in the private cloud or have them federated on-demand to partner-hosted public clouds. **vCloud** is providing tough competition to the more established AWS by Amazon.

3Tera

Offers **CloudWare:** an architecture to provide an open framework to allow the development of a cloud computing environment that is open enough to work on any web/enterprise application.

NetSuite

Specializes in SaaS. It offers **SuitCloud** Platform: a comprehensive offering of on-demand products, development tools and services designed to help customers and software developers take advantage of the significant benefits of cloud computing. Also a leading provider of web-based *Business Software Suite* for CRM, ERP tools and Accounting.

IBM

IBM offers**SmartCloud.** IBM SmartCloud includes (IaaS), (SaaS) and (PaaS) offered through public, private and hybrid cloud delivery models. All offerings are designed for business use.

Joyent

Joyent is a cloud provider company to deliver all three layers of the Cloud Stack i.e. Cloud service for all 3 *aaS.

Microsoft

Specializes in PaaS. It offers **Azure**, a Windows-as-a-service platform consisting of the operating system and developer services that can be used to build and enhance Web-hosted applications. Azure is available for VS 2008 via Web Platform Installer.**Visual Studio 2010** has all the features to code, debug, and deploy a cloud service.

Salesforce.com

A leader in SaaS. Itoffers **Salesforce** CRM (Sales Cloud 2, Service Cloud 2) & the**Force.com** Platform (Custom Cloud 2, Development Platform).

Rackspace

The company specializes in IaaS. It offers **Rackspace Cloud**, which includes Cloud Sites, Cloud Servers, and Cloud Files.

5 Solutionsfrom Developers

5.1 Google Applications

The Google web browser Chrome offers significant potential for third party applications to facilitate cloud computing.

Google focuses on the development of three major directions [1]:

- to improve Web search functions and enrich it with new possibilities (geo tagging, access to local resources, etc);
- development of system-platform tool that is mainly a web browser Chrome, the Android operating system for mobile devices and operating

system OS for netbooks Chrome. Chrome is to be equipped with the latest web standards implementations of HTML 5 in particular along with the native support for multimedia and networking and programming (AJAX, JSON, REST). Compliance with standards is needed, but the standards are formed in the trials and it makes the company ahead of the standards in order to introduce their products new functionality.

- construction of a usable web applications that run in a web browser and using the latest technologies and standards (HTML 5, AJAX, JSON, REST).

Google actively encourages the use of its products by end users, but also helps developers in the use and incorporation of modules into their own solutions to Google. It does this by providing a turnkey Web-based (closed components) to be built on your own website and by providing APIs (application program interface), which is invisible to the induction of components of the application.This company offers a number of groups of applications and components for different groups of Internet users, from end-users only viewers of web resources, web masters, by creating the web sites up to programmers developing web applications. A common feature of Google services and applications is to run these applications in a web browser. All the code that contains the application logic and presentation layer is generated on Google's servers and sent to the browser in the form of a specially crafted HTML pages. These pages are embedded tags <script> what's the JavaScript code that runs when the page is loaded into the working memory browser.

The main burden of the application resides on a server somewhere in the Google cloud. Any computer on any operating system equipped with a browser can launch the Google and get access to data stored in databases in the cloud. Google Apps are created with free tools. This means huge savings for the purchase of software. Google programmers program primarily in Java, JavaScript and Python. To increase the efficiency of creating the applications it is used the Google Web Toolkit (GWT).

The web browser Chrome is a new phenomenon in the world of third party applications. So it's not just the browser, but a fully-fledged tool (the container) to create web applications web 2.0

In order to create applications in the cloud, three options are available on Google Chrome:

- Plug-ins - is nothing but an opportunity to call own page through a shortcut on the browser toolbar.
- In applications, Google Docs (called Google Apps) can now be programmed using JavaScript and a special library of Google - the technique is called Google Apps Script.

We get the Google portal in two versions: the basic case of anonymous access (with a window to enter search terms) and expanded to registered users (here we have access to a personalized portal - user dashboard) also in two versions: a modest (services are listed on the bar at the top of the screen at the browser) and rich (containing googlets - ie mini portals to sites that you previously chose). In

this way, without installing any plug-ins every Google user can build a home page. Data on what services and what are the chosen portals are stored by Google in the user database in the cloud

5.2 Microsoft Applications

The most relevant of Microsoft's technological innovations for cloud computing include Silverlight technology and Azure , the cloud processing platform. Silverlight allows the creation of multimedia applications in a web browser.

Microsoft Azure, as depicted in figure 5, is a platform for cloud computing.The concept is based on virtualization of resources through virtualization software. The service for user applications consists of four components:

- Windows Azure - a service that can install applications on Microsoft servers. It is used to run the final applications written for processing in the cloud.
- SQL Azure - a database located in the cloud. It provides T-SQL language for developers who write applications in a cloud.
- Azure AppFabric - its role is to act as a gateway or router between objects on the Azure platform and objects along with their local network infrastructure and Active Directory authentications and certificates. This service also opens the final address in the HTTP or REST-style, making it available for applications outside environment of .NET. Azure Fabric acts as a wrapper for existing applications, configuration, instead of demanding substantial changes to the application code.
- "Dallas" –this is a typical service for solution providers in the cloud. It provides a directory of applications that are designed specifically to work in the cloud.

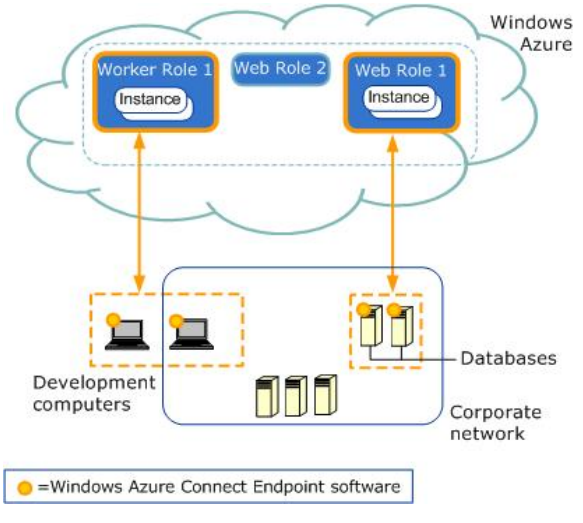


Fig. 5 Windows Azure [17]

The most important aspect of security applications in the cloud is to identify the object (person or process), which tries to get to the data in the cloud. Frequently we are dealing with the exchange of data between the client application (e.g., immersed in a web browser) and applications running in the cloud. For this purpose, a service .NET Access Control Service is used, which works with network services (web services) and supports many popular vendors of third-party credentials. Applications running in the cloud determine whether the user has access to a resource by examining a special token (SAML format is selected) created by a site called the STS (Security Token Service). This token certifies the identity of the user, in order to authenticate the certificate is signed by STS.

Microsoft Azure is subject to annual audit for compliance withPCI Data Security Standard (DSS), Sarbox (SOX) and Health Insurance Portability and Accountability Act(HIPAA) and constant internal control. In addition, MS has been certified to ISO / IEC 27001:2005 and SAS 70 Type 1 and II [18]This certificate refers to compliance with the requirements standard for the creation, implementation, processing, inspection, maintenance and improvement of safety management systems information stored in documents in the context of business security.

Security infrastructure clouds are classified according to level of security, and what they require (based on the risk of damage). Very sensitive resources are protected by powerful mechanisms such as multiphase authentication (biometrics, smart cards, and hardware tokens). There is a principle of preference, i.e. the lowest level of the person or the process receives the lowest level of preference for a particular resource, (but sufficient to handle the request for access.).

5.3 User's Rights of Applications in the Clouds

At the "Global Gartner IT Cloud Services Council for Rights and Responsibilities for Cloud Computing Services," conference Gartner presented a list of consumer rights and obligations important when choosing solutions provider in the cloud.[19]For services in the CC to become widespread the provider must offer guarantees to its proper operation.

1. The right to maintain full control over their data in the ownership, use, privacy, sharing, and access to their data
2. The right to amend the SLA containing responsibility for any failures and leaks information as well as methods of response and action in the event of failure and other abnormalities.
3. The right to early notification of planned downtime at work, monitoring the quality of services and the choice by the client before major events and changes in service.
4. The right to information on the scope, technical capabilities and limitations
5. The right to information about legal aspects in the local area where data is stored, and information about rights and obligations in the light of local law

6. Right to know the safety procedures and practices used by the provider to know all the strengths and weaknesses of their data security
7. The responsibility to review the rules of service and legal requirements arising from the use of licensed software

6 Summary

This review demonstrated the real possibility of creating a basis of free applications in the clouds, completely functional websites for groups supporting the work of an individual (planning, processing documents, e-mail) and in collaboration with others on the basis of available documents, calendar and e-mail exchange.

Processing in the clouds allows a new approach to client applications that run on MS Windows. A user can completely opt out of local data storage resources on the client computer. In this way we get rid of not only the installation and configuration problems associated with drivers for databases, but will create a new way of distributing the application code.

References

[1] Armbrust, M., et al.: Above the clouds: A Berkeley view of cloud computing. Tech. Rep. UCB/EECS-2009-28, EECS Department, U.C. Berkeley (February 2009)
[2] de Haaff, B.: Cloud computing - the jargon is back! Cloud Computing Journal. Electronic Magazine (August 2008), http://cloudcomputing.sys-con.com/node/613070
[3] Hand, E.: Head in the clouds. Nature (449), 963 (2007)
[4] Fink, J.: FBI agents raid Dallas computer business (April 2009), http://cbs11tv.com/local/Core.IP.Networks.2.974706.html
[5] Gray, J., Patterson, D.: A conversation with Jim Gray. ACM Queue 1(4), 8–17 (2003)
[6] Jackson, T.: We feel your pain, and we're sorry (August 2008), http://gmailblog.blogspot.com/2008/08/wefeel-your-pain-and-were-sorry.html
[7] Delic, K.A., Walker, M.A.: Emergence of the academic computing clouds. ACM Ubiquity (31) (2008)
[8] Krebs, B.: Amazon: Hey spammers, Get off my cloud! Washington Post (July 1, 2008)
[9] Fogarty, K.: CIO.com (accessed on October 6, 2011)
[10] NIST Information Technology Laboratory, http://www.nist.gov/itl/csd/cloud-102511.cfm
[11] Why Cloud Computing?, http://www.appistry.com/cloud-info-center
[12] Bigelow, S.: How server consolidation can benefit your data center, http://searchservervirtualization.techtarget.com/tip/How-server-consolidation-can-benefit-your-data-center

[13] Johnston, S.: Cloud Computing Types: Public Cloud, Hybrid Cloud, PrivateCloud (2009), http://www.circleid.com/posts/20090306_cloud_computing_types_public_hybrid_private/
[14] http://www.opengroup.org/cloud/whitepapers/ccroi/intro.htm
[15] http://www.computerworld.com.au/article/401509/australian_cloud_services_revenue_set_dramatic_growth_idc/
[16] http://www.techno-pulse.com/2009/12/top-cloud-computing-service-providers.html
[17] http://msdn.microsoft.com/en-us/library/windowsazure/gg432997.aspx
[18] http://sas70.com/sas70_overview.html
[19] http://www.gartner.com/it/page.jsp?id=1398913
[20] http://www.webgranth.com/a-complete-reference-to-cloud-computing

Design of Mobile Software Architecture

Ji-Uoo Tak, Roger Y. Lee, and Haeng-Kon Kim*

Abstract. Mobile software development challenge the modelling activities that precede the technical design of a software system. The context of a mobile system includes a broad spectrum of technical, physical, social and organizational aspects. Some of these aspects need to be built into the software. Selecting the aspects that are needed is becoming increasingly more complex with mobile systems than we have previously seen with more traditional information systems. With great diversification in the software market a mobile embedded system is loaded with software from dozens of different vendors. With the wide variety of different mobile embedded systems applications, we need specific software for those specific mobile embedded systems that will meet the requirements of an application. In this paper, we the design and implements the procedures for mobile software architecture using Components Based Development(CBD) and object oriented methodology. It starts with the requirement analysis of a mobile embedded systems and continues to provide the functional model of the system and also the environmental design as well. As more and more functions or components needed to be added in designing a software system a modularized and systematic approach becomes not an option. This paper proposes a systematic and procedural approach that will help build a more reliable and suitable mobile software architecture.

Keywords: Mobile, Software Architecture, Component based Development, Component discovery and notational representation, Architectural design.

Ji-Uoo Tak · Haeng-Kon Kim
Department of Computer Engineering, Catholic University of Daegu, Korea
e-mail: lebbenle@cu.ac.kr, hangkon@cu.ac.kr

Roger Y. Lee
Software Engineering Information Technology Institute,
Central Michigan University, USA
e-mail: lee1ry@cmich.edu

* Corresponding author.

R. Lee (Ed.): Software and Network Engineering, SCI 413, pp. 133–146.
springerlink.com

1 Introduction

Developments in the area of software architecture over the past two decades have pushed architecture to the forefront of a number of critical software engineering activities: modeling, design, analysis, simulation, implementation, deployment, maintenance, and evolution. Mobile software architecture is advocated as an effective conceptual tool for addressing the many challenges of developing large, complex and mobile software systems. Largely in parallel to these developments, significant advances have also been made in the domains of mobile, autonomic, service-oriented, grid-based, and most recently, cloud-based computing. The systems in these domains are also large, complex, and distributed; they are frequently expected to dynamically adapt to failures as well as changing requirements and execution contexts. At first blush, a number of the advances in these domains appear not to have resulted from an explicit software architectural focus. However, we posit that architecture offers clear, and often critical, benefits in these domains. In support of this argument, we will overview the state-of-the-art in the areas of mobile, autonomic, service-oriented, grid-based, and cloud-based computing, with a specific focus on the role software architecture should and actually does play in these domains[1,2,3]. We will highlight the characteristics of software architectures as well as specific architecture-based approaches that make them particularly suitable to developing these systems. Software architecture is the high-level structure of a software system or the organizational structure of a system. Architecture can be recursively decomposed into parts that interact through interfaces, relationships that connect parts, and constraints for assembling parts. Every software architecture should follow a specific methodology. There are many different software methodologies.

Hardware and software are both important means for mobile software system design and implementation. Mobile software system development process begins by making hardware and a software specification. The growing consumer demands for more functionality tools have lead to an increase in complexity of the final implementation of such design. However, even though current IC technology is following the growing consumer demands, the effort needed in modeling, simulating, and validating such designs is adversely affected. This is because current modeling method and frameworks, hardware and software co-design environment, do not fit with the rising demands[4,5].

In this paper, we discuss the software architecture for mobile embedded systems and its behaviors. The objective of explaining what mobile embedded software systems are is followed by presenting some of the most common definitions of software architecture. To start with mobile embedded software systems are those which need the results within definite timescales, otherwise the system just won't work properly. By the word timescales, we mean the system is bound by specific deadlines for the task to be finished. The basic component of a mobile software systems consist of a controlling system and a controlled system. A controlled system can be any real time environment as to be monitored by computer software. This paper describes mobile software architectures as those

systems with time-bounded response constraints. Other mobile embedded systems include command and control system, process control system, knowledge base, and multimedia and high speed communication systems. The aim of the work in this paper is to present a new radical approach for building software architecture for mobile embedded systems. Defining software architecture would help us understand the future literature of this paper.

2 Related Works

The development process of software architecture covers a set of activities whose nature and ordering depend on the particular system, on the designers' skills, and on the tools available. Some of these activities are performed by hand while dedicated tools support others. In any case, the key activities in software architectural design include: functional and modular decompositions, functions allocation to modules, processes identification, mapping modules with processes and mapping processes with processors. These activities bring to bear architectural design guides such as software architecture reference models, architecture styles and patterns, and result in the production of a conceptual architecture followed by an implementation architecture. Figure 1 shows the design process for software architecture ranging from reference models to implementation architecture.

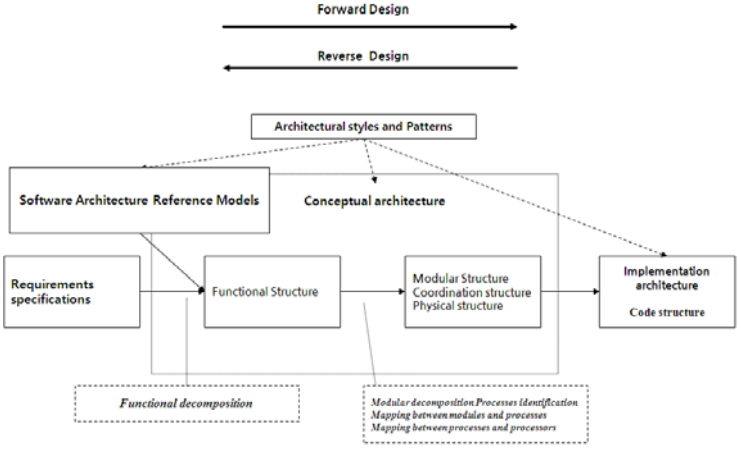


Fig. 1 Software architecture reference models

The *functional decomposition* of a system consists in expressing the functional requirements of the system into a set of simpler units. The components of this structure are abstraction units that express the functional requirements of the system. Their relations are of the type "exchange data with".

A *software architecture reference model* is a standard decomposition of known systems into functional components coupled in a well-defined way. For example,

a compiler is decomposed into four (possibly five) successive well-defined functions: the lexical, the syntactic, the semantic (and the pragmatic) analyses, followed by code generation.

The *modular decomposition* results in the definition of a static view of the system: the *modular structure*. The components of the structure are modules linked by relations such as "is a sub-module-of". The decomposition is performed according to software engineering rules and properties such as module dependency minimization. Based on the modular structure, the project manager can dispatch the development of the system to different programming teams. *Function allocation* is closely related to modular decomposition. It consists of allocating the functions of the functional structure to the modules of the modular structure. In other words, this activity results in the definition of the functional coverage of each module. A module may cover multiple functions. Conversely, a function may be distributed across multiple modules[6,7].

Formulating a general methodology to the development of mobile embedded systems must meet the high performance, size constraints and functionality. Many methodologies have been proposed before and most of them have focused on representations that used graphical notations, particularly flow charts. To inculcate a more structured design, a methodology which represented the system into functions and interfaces in communication between each module became more prominence. To represent the behavioral characteristics finite state machines were used. Module cohesion and module coupling are used to evaluate the methodology. Module cohesion is used for identifying the strength or unity within the modules and module coupling were used to determine the degree of connectivity between modules.

An alternative approach was also designed in which the emphasis is mode on first designing the program structures based on the data structures. Jackson system development is an instance of the above methodology. In this the design is based on modeling the reality at first before any functions of the system is being considered. This approach is much like a simulation and each is modeled by means of a concurrent task called a model task. This desing methodology used entity structure diagrams rather than individual symbols.

Finally the object-oriented methodology including [8] gained much importance and software design industry gained momentum.CBD and object oriented design used abstraction to separate object specification form its body. Information hiding is a major aspect o this method. In this paper, we take an object-oriented methodology that will help software designers to built efficient mobile embedded systems.

3 Functional Model of a Mobile Software Architecture

As the general complexity of the system increases so does the development complexity of the mobile embedded systems. Building software architecture for a mobile embedded systems needs a precise understanding of what exactly a

proposed system should do. This is the fundamental purpose of the analysis and specification – the requirements phase.

Specification requirements are exactly where start our framework for the architecture. This paper first aims at identifying the key functions of any mobile embedded systems. These functions are then modeled and defined precisely with different language tools. This paper as its first step identifies some of the most common and widely known requirements of a mobile embedded system. The system must provide a way to sense the real world changes and convert them into information. The system must have hard-fast/soft-slow design and the system should give the user the required information not after the deadline If hard-fast, then exactly before the deadline and If slow-soft the system may be time bound but still should not take more than some few seconds after the deadline. The system should process and validate the information sent by the external devices. The system must have a repository to store the processed data. It must provide a robust means of controlling the flow of data. Figure 2 shows the our mobile general frameworks and architecture model. It sonsists as Java virtual model, JavaFX Platform,Tools and applications with contents and services.

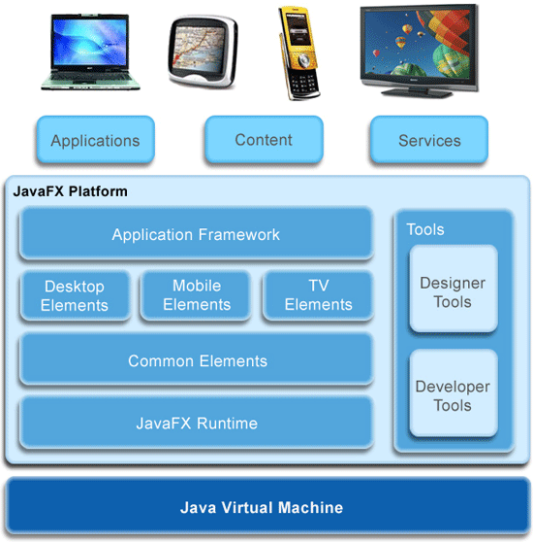


Fig. 2 Mobile Software architecture model in our works

3.1 Mobile Software Functional Design

With these formal specifications we can build a functional view of the system. The functional view of a real-time system helps us identify precisely what each function or a module is supposed to do once they are integrated within the system. Once these functions are clear then building a more complex and complete

real-time system would not be a difficult task. The below figure shows the pictorial representation of the functions a mobile software system will perform under a given condition as in figure 3.

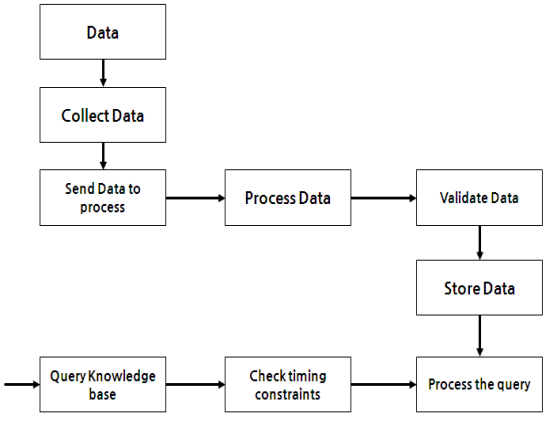


Fig. 3 Functional view of mobile software

This paper aims at representing architecture for real tiem systems using an component based development and Object-oriented methodology. Before we could present an architecture let's see what exactly a functional view represents. As we all know from the definition of mobile embedded systems that every mobile embedded systems should have some kind of data do precess from the controlled system and use that data to control back the objects of the real world.

The function of the first block is to collect the data from the real world. This data is then processed and should vevalidated. The validation is a more significant aspect of mobile embedded systems as these processed data are those which are used to control back the objecets in mobile embedded systems. Once the data is validated it is then stored in a repository for future retrieval or for a query process. This explanation briefs only the basic functions that a system must perform to be called a real-time system. One can always branch out and find more specific functions depending on for what a real-time system is used. Also identifying and defining each and every possible function for a real-time system is not feasible and is out of the scope of this paper. Given a real-time system formulting its functions is up to the audience of this paper.

3.2 Mobile Software Environmental Desing

This section explains a development environment in figure 4 for the software designers which will help them design better and more efficient software architecture for mobile embedded systems. This environment fives the user a graphical view of the architecture built using the basic notations summarized later

in this paper. Since most of the mobile embedded systems communicate with the objects in the outer world an Object-oriented approach suits well with the world as the world deals only with objects. Also objects have its own communication patterns and behavioral patterns.

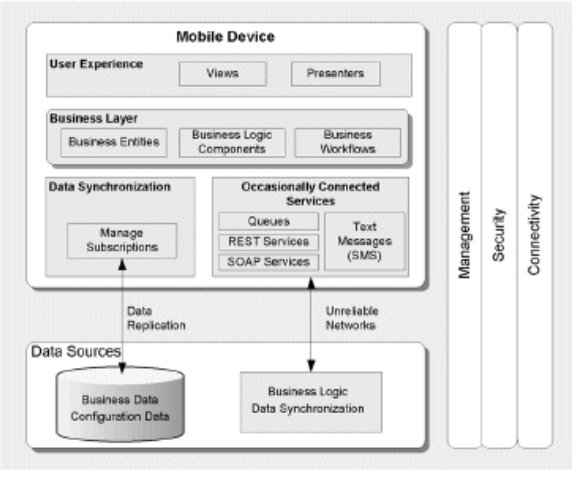


Fig. 4 Environmental architecture for mobile embedded software system

The environment for mobile embedded software architecture is best understood with the design. As said earlier formulating a specification requirement for the system we propose to build will be a key aspect to begin with. Development environment then classifies the specification requirements into embedded system specification requirement and software specification. This helps designers to simply the job of identifying components and its functions. Hence this paper highly values object-oriented approach at all levels. All mobile embedded systems should have some form of embedded system and that system will vary depending on task the system is being built. For instance a smoke detector is a simple mobile embedded systems containing embedded system with one micro controller but on the other hand a large real time knowledge system should have a more efficient multi processor with memory at least double the conventional mobile embedded systems.

3.3 Specification Analyzer

The embedded system specification analyzer module evaluates the modeled architecture to check whether all the specification requirements have been met. A repository module stores the modeled system for future use. Though at this time there are no notations used the paper will be using the basic notations to design the architectural details of the system. Every device is represented as object and each

object is given its own attributes and behaviors. Also association between each object is represented using the notations given in this paper. The object in the system communicates using specific signal types.

3.4 Elements of Mobile Software Architecture

A notational diagram and its semantics are stored in a file for the designers to use. These descriptions have both the semantic and syntactic details for the system. For example, one may use function names like call() that will be used as a procedural call. It's syntactic details consists of how and where a call should be used and in what format while semantic details consists of what the function does i.e., calling another function or subroutine. The notations for software architecture describe the system graphically. These notations present information about the system in a simple and a more comprehensive manner.

Before the software architecture notations are described it is essential to explain some of the most significant aspects of the software architecture. Some of the most common elements to be explained are Packages, functionality, quality requirements and architectural styles. The explanation for the software architecture elements presented in this paper is form various research groups and research papers.

Every notation is represented by both components and connectors. The components are the real world entities in the system or a device in a system. The connectors are the interfaces between two components and they define the relationship between two components.

The components having same property or those components behaving similar are grouped into same modules. Those components that are grouped together form packages. Each notation has its own representation of packages and components. This paper uses the basic concepts of Object-oriented approach to represent a component.

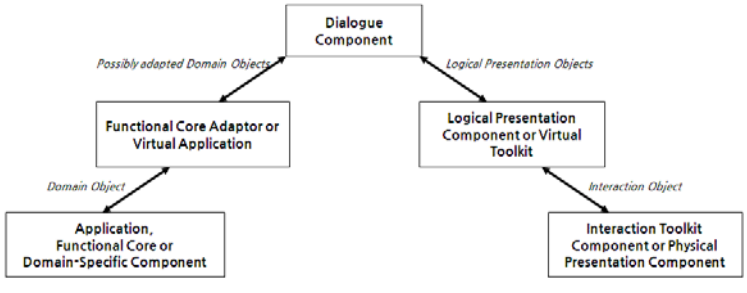


Fig. 5 Elementsof mobile software architectures

As in figure 5, functionality is another important element of software architecture. The functionality of a system represents the behavior of a system at a specific time. This is usually implemented at the programming level. Quality requirement is another aspect which deals with the robustness, security, scalability and other performance aspects.

Finally architectural styles gives the designers to choose between different styles available based on the type of data flow. The following examples are borrowed from[7]. These components are used in the design of the software architecture. The environment presented in this paper provides the embedded system architecture suitable for the specified application and the system software architecture as well[10].

3.5 Mobile Software Model

In this framework each module processes an event and the result of each event is sent to the appropriate components. In this section we describe object orientation followed by the software model designed using object-orientation. The architecture design diagram for mobile embedded systems is shown in figure 6. This architectural design uses CBD and object-oriented approach. Also this paper represents each object with its own notation so that the designers will not be confused when they use too many notations. Though the design seems to resemble layered approach the object-oriented concepts is been largely used.

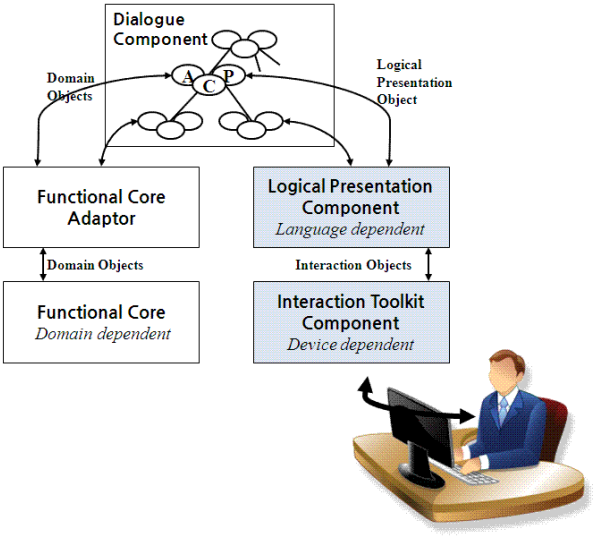


Fig. 6 Mobile Architectural Design process

(1) Software Model

All mobile embedded systems use some kind of external device to communicate to the real world. These are called sensors and actuators. These objects are represented using nodes, the basic component defined before. This node should be able to explain to attributes and its behavior at a particular state. Each component given in the figure 6 contains a function that performs a specific action. Once these functions are executed with no errors components produce a set of intermediate result or final result.

(2) Components

One components' output may be an input for another component. In such situation the component which produced a result is referred to as *producer agent* and later is referred to as *successor agent*. These can also act as either a passive object or active object. In this framework a *State manager object* maintains the status of a passive and an active object. This helps to maintain a record of which component is currently producing result and which is consuming. Also with the use of state manager object we can maintain a record of each object's status and function description. These records are then stored in the repository operated by Database Management module. Any alteration for a passive object needs only one change at the central database. Hence redundancy is not an element to be questioned.

(3) Component Interaction

Each basic building of the system is connected using other objects called connectors. These connectors are responsible for the data transmission between components and devices. Each time a data is transmitted from one component to another the component which sent the data is represented as SEND component and the component which received is represented as RECEIVE component. This information is attached along with the data object being sent. Each time a data is being received it is checked for the validity by Data Analysis module.

(4) Mobile Embedded System Architecture

This is an important module in mobile embedded systems and is carefully designed depending on many constraints and specifications defined earlier. The environment development unit clearly separates and identifies the embedded system specification from the software specification with the intention of making designers job simple. We all know that an embedded system should contain one or more microcontrollers to perform some action. This architectural design assumes that most of the mobile embedded systems that are built now need more than one microcontroller as today's necessity has increased largely. These microcontrollers are again represented as individual objects. This diagram is a functional representation of the mobile embedded systems. The data flow in a mobile embedded systems is represented using notational connectors.

4 Design Mobile Software Architecture

A good understanding of the system specification and analysis of that specification is the key activity in requirements analysis phase. In conceptual design phase a high-level design of the system is built. In this phase the system design will not contain the detailed architecture but rather an overall structure. This is an abstraction level design where most significant object and their behaviors are identified ignoring other unimportant objects.

Once we have the conceptual design we now have to define the functions of each object in the mobile embedded systems. In this phase a pictorial

representation of the functions or tasks carried out at each stage of the system is provided. This functional view will be of great use for the designers when they reach the implementation phase of the design. Before we start designing a bule print for a system we needto identify what objects or external devices should be used based on the functional view developed in the previous phase. This is done in the object discovery process where we highlight all the nouns or noun phrases in the problem domain.

In this process the objects of interest are first identified and then the object of least interest are highlighted. Once we find the objects of interest we need to classify them into active and passive objects. Active objects are those objects that requests services from a passive object. Passive objects can be considered as servers.

This phase also includes identifying the physical devices that should be used in a mobile embedded systems. Also identifying the key concept for using an object at this stage will largely reduce the problem in the future.

This phase builds a descriptive model of the high-level design we provided before. This descriptive model expands the high-level design into a more detailed description of the system. This descriptive model helps to identify functions and partitioning of subsystems, type of interfaces between objects, application of specific solution to sub problems etc.,

Modularization is one of the important aspects in component and object-oriented approach where one large problem is being broken down to many sub problems. This way we reduce the complexity of the problem largely. This is achieved by first identifying the boundaries of each objects and how they interact with other objects. Once we are clear about these details then we can break down a big module into chunks of smaller mocule. Each problem is then handled by different person.

This is the final phase of the guidelines for building mobile embedded systems. There are many different kinds of testing which will not be discussed in this paper. Correntness, reliability and maintainability of the system should be tested. Also each individual unit or module should be tested for the interoperability and scalability issues as in figure 6.

4.1 Sample Mobile Architecture for Multimedia Mobile Conference Management Systems

We suggest and apply our mobile architectur to multimedia mobile conference management embedded systems that is a client/server based architecture with 3 main components

- Directory servers
- Conference Managers
- Conference Agents

These are a collection of conferencing system object. Directory servers provide user authentication and name-to-address mapping functions. Each directory server

is responsible for a domain, or position of a network, which consists of one or more conference managers. So these conferencing managers can be grouped into a package of directory servers. These directory servers can be altered to add or delete domain name or addresses in the directory servers.

Also any association, data flow and other user interfaces if present has to be defined at this stage. From the environment architecture the modeling Embedded system and software design modeling represents this. Any system needs to be modeled before it is built and also these models should be converted to a notational diagram. Thus these directory server, Conference managers and other components and connectors of this system are first modeled and represented using notations.

Using the basic notations we represent these directory servers as follows and if these directory servers contain one or more conference managers then it will be referred to as a Module Directory server.

These components establish, maintain and remove connections between conference participants. They are also responsible for sending multicast messages to the conference agents that will be defined in the nest section. These conference managers participate in the conference with other conference objects. Conference managers are also represented using component objects.

Mobile architectur for multimedia mobile conference management embedded systems is represent using UML as in figure 7. They are responsible for formatting and transmitting messages to their corresponding conference managers. These conference agents communicate with one another indirectly through their corresponding conference managers.

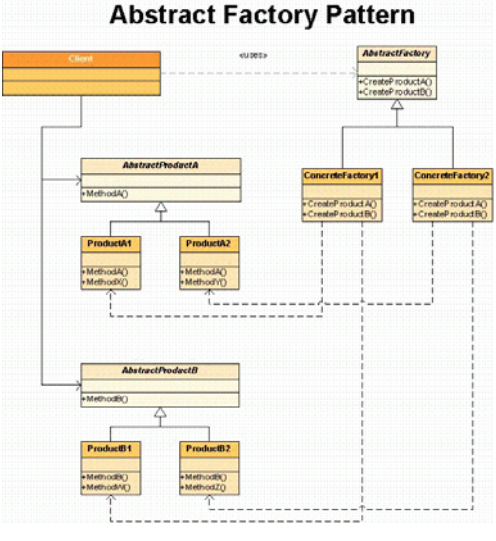


Fig. 7 UML Architecture Component Representation with Factory Pattern

4.2 Component Interaction

Once the objects of the system is defined identifying the active and passive objects are important. With the method provided identify the passive and active objects. These are represented in the notational diagram. Once all the objects and its components are defined we need to identify or present the interaction between these components in a clear way.

The component interaction gives overall understanding of the system as in figure 8. These are usually represented using connectors and this paper provides many different types of connectors based on type of data flow. Interaction of conference agent, conference managers and directory servers varies depending on the state of a conference. For example, when scheduling a conference only directory severs and conference managers anre involved. When a conference manager receives a request to schedule a conference, it interacts with its local directory servers to obtain location of the invited participants.

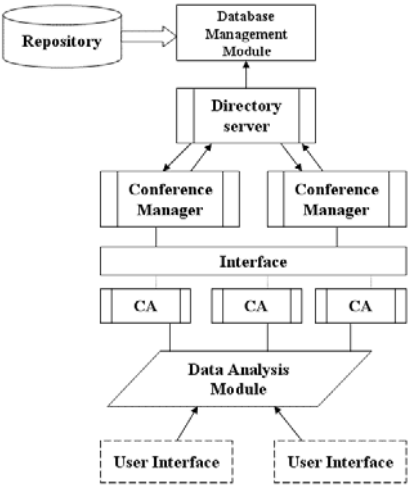


Fig. 8 Mobile Software Architecture for Multimedia system

5 Conclusion

Mobile software systems are becoming a pervasive element of society. Many methodologies for these systems have made a significant progress in near future. In this paper, we have contributed by presenting a new guidelines and general notations for building mobile embedded software systems. This paper used componnet based development and object-oriented approach to both identify and define different types of objects and their interfaces. We exploited the CBD and Object-oriented concepts to build a new set of notations which is then used to design an architectural structure. The guidelines and procedures for building

architectural design in form of a flow chart format has been used to show that a new architecture can be designed. In the futre we are are going to apply it to many different application area as a presence of the case studies, the paper is complete as it proves the validity of the software architecture. With this background we intend to work on the answers to some of the most ambiguous questions that may arise in the near future and to implement an environment where we can use procedures and notations to formulate architecture for specific mobile embedded systems.

References

1. Nakagawa, E.Y., Ferrari, F.C., Sasaki, M.M.F., Maldonado, J.C.: An aspect-oriented reference architecture for Software Engineering Environments. Journal of Systems and Software 84(10), 1670–1684 (2011)
2. Weinreich, R., Buchgeher, G.: Towards supporting the software architecture life cycle, Procedia Computer Science. Journal of Systems and Software (available online June 7, 2011) (in press, corrected proof)
3. Breivold, H.P., Crnkovic, I., Larsson, M.: A systematic review of software architecture evolution research. Information and Software Technology 54(1), 16–40 (2012)
4. Asadollahi, Y., Rafe, V., Asadollahi, S., Assadollahi, S.: A formal framework to model and validate event-based software architecture. Procedia Computer Science 3, 961–966 (2011)
5. Neyem, A., Ochoa, S.F., Pino, J.A., Franco, R.D.: A reusable structural design for mobile collaborative applications. Journal of Systems and Software (available online June 7, 2011) (in press, corrected proof)
6. Gavalas, D., Bellavista, P., Cao, J., Issarny, V.: Mobile applications: Status and trends. Journal of Systems and Software 84(11), 1823–1826 (2011)
7. Kovač, D., Trček, D.: Qualitative trust modeling in SOA. Journal of Systems Architecture 55(4), 255–263 (2009)
8. Hartmann, H., Trew, T., Bosch, J.: The changing industry structure of software development for consumer electronics and its consequences for software architectures. Journal of Systems and Software 85(1), 178–192 (2012)
9. Hsueh, N.-L., Shen, W.-H., Yang, Z.-W., Yang, D.-L.: Applying UML and software simulation for process definition, verification, and validation. Information and Software Technology 50(9-10), 897–911 (2008)
10. Xie, F., Yang, G., Song, X.: Software complexity and its impacts in embedded intelligent real-time systems. Journal of Systems and Software 78(2), 128–145 (2005)
11. Meedeniya, I., Buhnova, B., Aleti, A., Grunske, L.: Reliability-driven deployment optimization for embedded systems. Journal of Systems and Software 84(5), 835–846 (2011)

Author Index