

Chief Editor : Dr. Ruben Prieto-Diaz



CONTENTS

Vol. 1 No. 1

December 2008

Eliciting Theory about a Retirement Process	
M. Kajko-Mattsson, A. Hauzenberger & R. Fredriksson.....	1
A Bioinformatics-Inspired Adaptation to Ukkonen's Edit Distance Calculating Algorithm and Its Applicability Towards Distributed Data Mining	
B. Johnson.....	8
Designing and Verifying Communication Protocols Using Model Driven Architecture and Spin Model Checker	
P. S. Kaliappan & H. Koenig.....	13
A New Communication Framework for Networked Mobile Games	
C. W. Xu.....	20
Storing and Searching Metadata for Digital Broadcasting on Set-Top Box Environments	
J. H. Park & J. H. Kang.....	26
Development of an Improved GUI Automation Test System Based on Event-Flow Graph	
Y. Z. Lu, D. P. Yan, S. L. Nie & C. Wang.....	38
An Evaluation Approach of Subjective Trust Based on Cloud Model	
S. X. Wang, L. Zhang, N. Ma & S. Wang.....	44
Motif-based Classification in Journal Citation Networks	
W. C. Wu, Y. N. Han & D. Y. Li.....	53
Two-Tier GCT Based Approach for Attack Detection	
Z. W. Wang, Q. Xia & K. Lu.....	60
Towards Automatic Transformation from UML Model to FSM Model for Web Applications	
X. Wang, H. K. Miao & L. Guo.....	68
An Algorithm for Generation of Attack Signatures Based on Sequences Alignment	
N. Li, C. H. Xia, Y. Yang & H. Q. Wang.....	76
Workflow Mining of More Perspectives of Workflow	
P. Liu, B. S. Zhou.....	83
Complying with Coding Standards or Retaining Programming Style: A Quality Outlook at Source Code Level	
Y. Q. Wang, B. Zheng & H. J. Huang.....	88

Journal of Software Engineering and Applications (JSEA)

Journal Information

SUBSCRIPTIONS

The *Journal of Software Engineering and Applications* (Online at Scientific Research Publishing, www.SciRP.org) is published quarterly by Scientific Research Publishing, Inc. 5005 Paseo Segovia, Irvine, CA 92603-3334, USA.

E-mail: jsea@scirp.org

Subscription rates: Volume 1 2008

Print: \$50 per copy.

Electronic: free, available on www.SciRP.org.

To subscribe, please contact Journals Subscriptions Department, E-mail: jsea@scirp.org

Sample copies: If you are interested in subscribing, you may obtain a free sample copy by contacting Scientific Research Publishing, Inc at the above address.

SERVICES

Advertisements

Advertisement Sales Department, E-mail: jsea@scirp.org

Reprints (minimum quantity 100 copies)

Reprints Co-ordinator, Scientific Research Publishing, Inc. 5005 Paseo Segovia, Irvine, CA 92603-3334, USA.

E-mail: jsea@scirp.org

COPYRIGHT

Copyright© 2008 Scientific Research Publishing, Inc.

All Rights Reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as described below, without the permission in writing of the Publisher.

Copying of articles is not permitted except for personal and internal use, to the extent permitted by national copyright law, or under the terms of a license issued by the national Reproduction Rights Organization.

Requests for permission for other kinds of copying, such as copying for general distribution, for advertising or promotional purposes, for creating new collective works or for resale, and other enquiries should be addressed to the Publisher.

Statements and opinions expressed in the articles and communications are those of the individual contributors and not the statements and opinion of Scientific Research Publishing, Inc. We assume no responsibility or liability for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained herein. We expressly disclaim any implied warranties of merchantability or fitness for a particular purpose. If expert assistance is required, the services of a competent professional person should be sought.

PRODUCTION INFORMATION

For manuscripts that have been accepted for publication, please contact:

E-mail: jsea@scirp.org

Eliciting Theory about a Retirement Process

Mira Kajko-Mattsson, Anna Hauzenberger, Ralf Fredriksson

Dept. of Computer and Systems Sciences, Stockholm University/Royal Institute of Technology/Karolinska Institutet Sweden
Email: mira@dsv.su.se

Received November 25th, 2008; revised November 28th, 2008; accepted November 30th, 2008.

ABSTRACT

The software community has been so much focused on creating and improving development and evolution processes, so that it has completely forgotten retirement. Today, there are no retirement process models whatsoever despite the fact that many software organizations desperately need guidelines for retiring their old software systems. In this paper, we elicit a retirement process model and compare it to the current retirement process models. Our goal is to educate theory about retirement process, evaluate current retirement process standards and provide feedback for their extension. The elicitation process has been made within one Nordic financial company.

Keywords: Archival, Conversion, Migration, Process Model

1. Introduction

Research on software lifecycle process models has not been well balanced so far. Most of the attention has been paid to software development and evolution. Less focus has been put on software maintenance. No research has been made on software retirement whatsoever.

Retirement is the disposal process whose aim is to end the existence of a software system [1]. It consists of the actual removal of a software system from a regular usage, migration of its still relevant parts to some other system(s) and the archiving of it [2].

There are plenty of reasons why a system needs to be retired. Some of them are the system age and complexity, removal of its software and/or hardware platform, rules embodied by the external environments, and the like. Irrespective of the underlying reasons, retirement is an extremely complex and difficult process. Hence, it must be carefully planned and performed.

Except for a very few standards, there are no retirement process models whatsoever. The extant standard models are not based on any real-life studies [3,4]. Due to the fact that their contents have mainly been chosen in ballots, they are very general. At their most, they cover a whole retirement process model within only a few pages. Hence, the current standards do not provide sufficient guidelines for the organizations in their complex retirement work.

In this paper, we elicit a retirement process model. Our goal is to provide a basis for creating theory on the domain of a retirement process, to evaluate current process standards and provide feedback for their extension. The elicitation process has been made within one Nordic financial company. This company has recently undergone two retirement projects, one in

Sweden and one in Finland. Both these projects were very comprehensive. They involved almost the whole organization and they took several years to complete. However, they differed in their prerequisites and process designs. For this reason, in this paper we only report on one of these retirement projects. The other project has been reported in [5]. The project reported herein is called EXIT and it was conducted in Sweden.

The remainder of this paper is structured as follows. Section 2 briefly presents the organization studied and the research method as applied in this study. Sections 3 and 4 describe the retirement process model as elicited within the organization studied. Section 5 compares our model to the existing retirement process models. Finally, Section 6 makes final remarks and suggestions for future work.

2. Research Method

This section describes the organization studied and the research method we followed when eliciting our model. Section 2.1 presents the organization and the systems to be retired. Section 2.2 briefly gives an account of our research steps.

2.1 Organization and Systems Studied

We studied one Nordic insurance organization. Due to the sensitivity of the results presented herein, we do not mention its name. Instead, we use its fictive name - FORSAK. FORSAK is the leading property and casualty insurance company in the Nordic region. It has about four million customers in the Nordic countries. It provides insurance services to both private customers and commercial and industrial organizations.

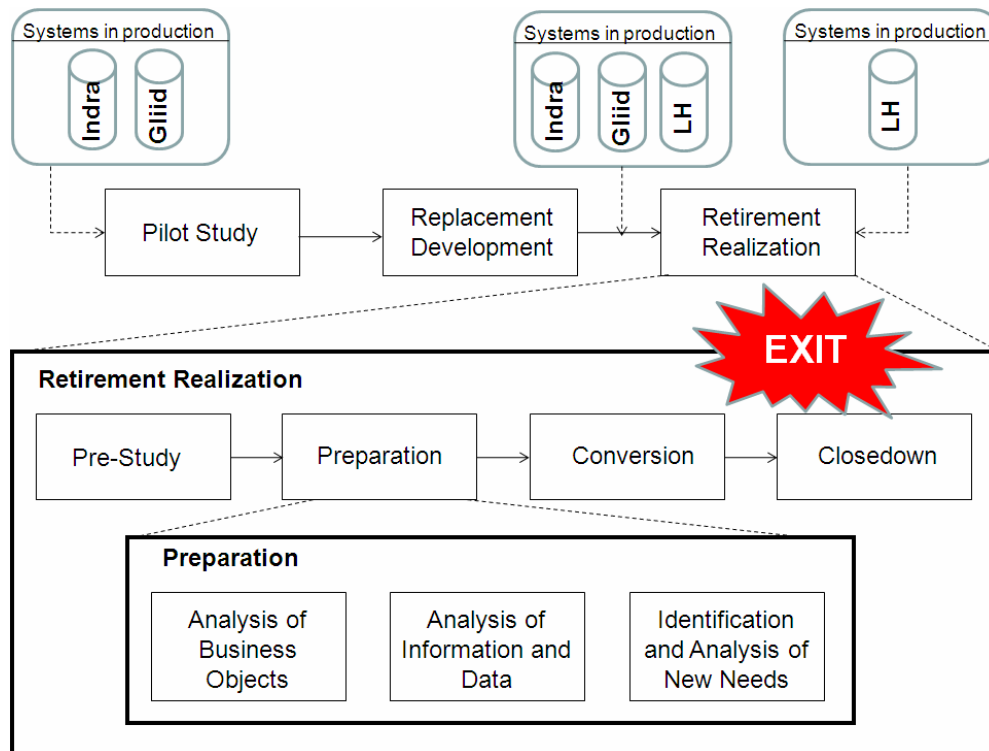


Figure 1. Retirement process phases in the EXIT project

FORSAK manages many systems. The systems that are of interest for this study are Indra and Gliid. At the beginning of the EXIT project (year 2002), Indra, based on a client-server architecture, was about 14 years old whereas Gliid, being a mainframe application, was about 20 years old. Both systems possessed overlapping functionality and were used by about 100-150 users.

The evolution and maintenance of Indra and Gliid as separate units was too expensive. First, it required substantially increased effort to implement the same functionality in two different systems. Second, the differences in their system designs forced one to conduct one and the same working routines in different ways. To avoid this, FORSAK has decided to take appropriate measures with respect to these two systems, that is, to retire them and replace them with a new system. The new system is called LH.

2.2 Research Steps

Our study was a typical design research [6]. Its goal was to explore a theory about and model the domain of retirement by identifying all its relevant process constituents and the relationships among them. It consisted of the following steps: (1) *Literature Study*, (2) *Study of the EXIT Project*, (3) *Creation of a Preliminary Retirement Process Model*, (4) *Model Evaluation*, (5) *Refinement of the Model*, and (6) *Comparison of the Model with the Standard Models*.

During the first step, we conducted an extensive and comprehensive literature study. We went through various articles and standard process models touching on the retirement subject. None of them, however, provided us with detailed information about the retirement process. Only [3,4] outlined very general process models. Due to their very coarse-grained nature, they did not provide any sufficient platform for starting our process design work. Hence, we may claim that the results of this work are entirely elicited from scratch using the industrial support.

In the second step, the *Study of the EXIT Project* step, we studied the EXIT project by first scrutinizing all the relevant project documentation. This documentation included about 100 different documents corresponding to retirement project descriptions, project plans, status reports, activity lists as created by individual project members, system overviews, reports from various meetings such as steering groups, reference groups, and the like.

The documents studied did not fully describe the whole retirement project. Hence, we had to complement our study with interviews. Here, we interviewed the roles such as a project manager, operation expert and decision maker.

Based on the understanding gained so far, we created a preliminary retirement process model. This model outlined a set of process activities in the EXIT project, it structured these activities into process phases, and it identified roles involved in them. This preliminary model

was then presented to the project manager in the organization studied. The goal was to evaluate its credibility and adherence to the EXIT project. The evaluation step resulted in some minor modifications to the EXIT retirement process model. These modifications are listed in Section 5.1.

Finally, we compared our model to the standard models [3,4]. To enable the comparison, we created a set of comparison criteria. Due to the fact that the standard process models studied are very general, we could define our criteria only on a very general level. These criteria are listed in Table 1. They mainly concern roles involved in and activities being part of the overall retirement process.

3. Overview of the Overall Retirement Process

The overall retirement process in this context consisted of three main phases. As illustrated in Figure 1, these were (1) *Pilot Study*, (2) *Replacement Implementation*, and (3) *Retirement Realization*.

During the *Pilot Study* phase in the year of 2002, FORSAK analyzed Indra and Gliid with the purpose of identifying cheaper solutions for managing these two systems. Two alternatives were suggested: (1) a merge of Indra and Gliid and (2) development of a replacing system LH and retirement of Indra and Gliid. The second alternative was chosen. It was regarded to be cheaper and more reliable in the long run. The *Pilot Study* phase ended up with a decision to start a project during which a new system LH would be developed and Indra and Gliid would be retired.

During the *Replacement Implementation* phase in the years of 2003 to 2005, FORSAK was in the process of developing LH. LH was developed in an iterative manner, where each iteration focused on a specific product domain, such as car insurance, home insurance, and the like. For this reason, the LH system was deployed in a successive manner in the years of 2005 and 2006.

After the new system was developed, FORSAK stepped into the *Retirement Realization* phase during which it disposed itself of Indra and Gliid. As illustrated by a star banner in Figure 1, this project was called EXIT. It took place in the years of 2006–2007. During this time, all the three systems were in use. In 2008, both Indra and Gliid were closed down and only LH has been used since then.

Table 1. Our comparison criteria

Roles
Activities
- System Analysis
- Archiving Strategy
- Migration Strategy
- Management of the adjacent systems
- Retirement planning
- Risk management
- Conduct archival

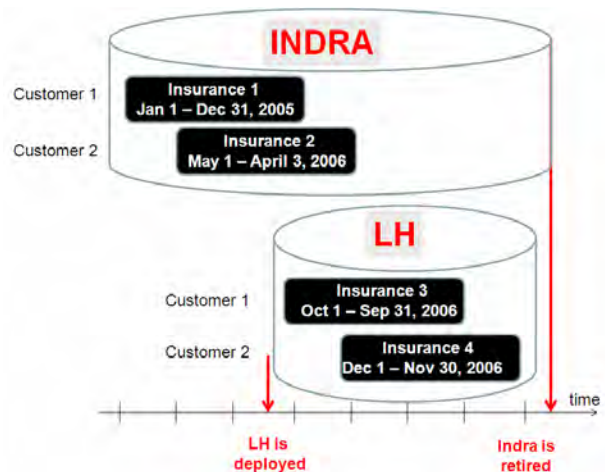


Figure 2. Illustrating the simultaneous administration of insurances in Indra and LH

Regarding the years 2005–2007, all the three systems were used in production. FORSAK was forced to keep the old systems running due to the fact that many of the insurances recorded in them were still valid. Indra and Gliid administrated all the old insurance cases whereas LH administered the new ones. This implies that insurances for one and the same customer were managed by the old and new systems simultaneously. The choice of the system to be managed at this time depended on the insurance period. Figure 2 provides a fictive example of how reported injuries for one and the same customer were administrated by two of the systems, Indra and LH.

4. The Project Phases

The EXIT project consisted of four phases are (1) *Pre-Study* (2) *Preparation*, (3) *Conversion*, and (4) *Closedown*. They involved the following roles:

- *System Manager (SM)*: a role responsible for the operation and maintenance of the system. *System Manager* manages the implementation, testing and deployment of all the prioritized change requests.
- *Decision Makers (DM)*: a set of managerial roles making important decisions within the organization. In the context of retirement, these roles may be project sponsors or managers of the departments affected by the retiring or replacing systems. *Decision Makers* are responsible for planning and managing the retirement process.
- *Operations Expert (OE)*: a role possessing expert knowledge of the organization's operation and the systems supporting the operation. *Operations Expert* also possess good knowledge of various rules and laws that may affect the retiring and/or replacing systems.
- *Project Leader (PL)*: a role that manages a retirement project. He plans, follows, and follows up the project. He also assures that right resources are assigned to the project.

- *User (U)*: a user of the systems to be retired. A user tests the results of the conversion of information and data from the retiring to the replacing system.
- *Developer (D)*: a role involved in the implementation of the retirement process. This role covers programmers, database developers and database administrators.
- *System Analyst (SA)*: a role responsible for planning and analyzing the software systems within the organization. He collects information and gathers requirements on the organization's operation, maps out its supporting processes and systems and the roles involved.
- *Support Technician (ST)*: a role responsible for the operation and support of the system to be retired and its software and hardware platforms.
- *System Architect (SAR)*: a role added to our model after the industrial evaluation step as described in Section 5.1. *System Architect* is responsible for knowing the overall architecture of the systems to be retired.

The EXIT retirement phases are illustrated in the box marked with a star banner in Figure 1. Their inherent activities and the roles performing them are listed in Table 2. Below, we describe each of the EXIT phases in Sections 4.1–4.4, respectively.

4.1 Pre-Study

The goal of the *Pre-study* phase was to investigate the systems to be retired, determine which of their parts should be migrated and disposed off, identify appropriate archiving and migration strategies, and define a retirement project and to plan for it.

As shown in Table 1, one first investigated the types of business objects managed by the systems to be retired. One then determined their volume. An example of a business object is an insurance, a customer or an encountered injury. Usually, the objects to be migrated were valid insurance claims.

Having an overall picture of the types and volume of the business objects to be migrated, one then determined the archiving and migration needs to be further used for identifying the appropriate migration and archiving strategies. However, no strategies were determined at this phase. FORSAK realized that deeper analysis was required for determining them. Hence, at this stage, one only determined that the active business objects should be migrated to the new system. Passive objects, on the other hand, should be archived. An example of an active object is a reported injury that has not yet been fully attended to.

After having identified the migration and archiving strategies, one determined the project scope. When doing it, one first analyzed Indra and Gliid's overall architecture and design. One then identified dependencies to other systems. Here, one considered systems and their users that were dependent on the retiring systems. Four interfacing systems were identified: two insurance administrative ones, one bookkeeping system, and one accounting system.

Identification of the interfacing systems affected by the closure of Indra and Gliid led to the identification of the additional activities required for managing the retirement project. In our case, one recognized (1) a need for analyzing the migration and archiving strategies, and (2) a need for making deeper analyses of the adjacent systems and their connections to the systems to be retired. These analyses were then conducted in the *Preparation* phase.

Finally, one defined a retirement project. The project definition included risk management and creation of a retirement plan. Risk management concerned risks such as access to resources required, staff illness and various technical risks. The retirement plan, on the other hand, covered most of the rudimentary project planning activities such as the identification of the stakeholders to be involved in the retirement project, identification of the roles required for managing and executing the project, determination of the competence required for managing and implementing the retirement process, determination of the project budget and schedule, and the like.

4.2 Preparation

The goal of the *Preparation* phase was to further analyze the systems to be retired, make a decision on archiving and migration strategies, determine changes to be made in the adjacent systems and to identify changes to be made in the replacing system.

As a first step, one studied the business objects to be migrated. The goal was to identify active objects and to attend to inconsistencies in them. To be able to recognize active objects, one had to define appropriate analysis activities and the roles required for performing them. An example of an analysis activity is a task to create a list of open injuries, go through the injuries and determine which of them should stay open and which of them should be closed. The open injuries were subjects for migration.

For all the active business objects, one analyzed their individual data fields in order to determine whether they should be migrated to the new system. One also analyzed special cases of business objects. An example of a special case is the situation when one and the same business object is administered by both the retiring and replacing systems.

For the data fields to be migrated, one created a conversion table so that the fields in Indra and Gliid would match the fields in the new LH system. Finally, one created a conversion testing plan. The testing implied that one chose a specific numeric field, summed it for all the business object instances in the systems to be retired and compared their sum to the corresponding sum in the new system.

As a next step, one determined the migration strategy. The choice was between manual and automatic conversion techniques. In total, one estimated that there were about 2400 active objects. To manually convert them would take about 20 man-minutes. This implied 100

Table 2. Retirement process phases and activities. The abbreviations in the parenthesis as listed after each activity correspond to the roles performing them. Underlined activities and roles written in bold text were added after the industrial process model evaluation step

<p style="text-align: center;"><u>Pre-Study</u></p> <div style="display: flex; justify-content: space-between;"> <div style="width: 48%;"> <ol style="list-style-type: none"> 1. Analyze the system to be retired <ol style="list-style-type: none"> 1. Estimate the extent of reuse (OE, SA, SM) <ol style="list-style-type: none"> 1. Identify types of business objects to be reused 2. Identify/determine the reuse volume (number of business object instances) 2. Identify archiving and reuse needs (BE, VF) 2. Identify the overall reuse/conversion strategy (DM, OE) 3. Identify the overall archiving strategy (DM, OE) 4. Determine the project scope <ol style="list-style-type: none"> 1. Analyze system architecture and design (SM, SAR) 2. Identify dependencies to other systems (SA, SAR) 3. Identify users affected by the system (SA, SAR) <ol style="list-style-type: none"> 1. Users of the retired system 2. Users of the systems dependent on the retired system 5. Identify other activities needed for managing the retirement (OE, SA) </div> <div style="width: 48%;"> <ol style="list-style-type: none"> 6. Define a retirement project <ol style="list-style-type: none"> 1. Manage risks <ol style="list-style-type: none"> 1. Identify risks (PL, OE, DM, SA, SM) 2. Analyze risks (PL, OE, DM, SA, SM) 3. Make a decision on how to manage risks (PL, OE, DM, SA, SM) 2. Create a Retirement Plan (DM, OE) <ol style="list-style-type: none"> 1. 2. Identify/determine stakeholders to be involved in the retirement project (PL, DM) 3. Identify/determine roles required for managing and executing the retirement process (PL) 4. Determine the competence required for managing the retirement (DM, OE) 5. Determine budget (PL, DM) 6. </div> </div>	
<p style="text-align: center;"><u>Preparation</u></p> <div style="display: flex; justify-content: space-between;"> <div style="width: 48%;"> <ol style="list-style-type: none"> 1. Analyze the business objects to be migrated/archived. For each type: <ol style="list-style-type: none"> 1. Define the analysis activities to be performed (OE, SM, SA) 2. Identify the roles to perform the analysis activities (OE) 3. Conduct the analysis (U) <ol style="list-style-type: none"> 1. Determine which business objects should be migrated/archived 4. Follow-up the analysis (OE) 2. Identify the laws and rules to be followed when migrating and archiving information and data (OE) 3. For each type of a business object: <ol style="list-style-type: none"> 1. Analyze how it should be converted (OE, SA) <ol style="list-style-type: none"> 1. Identify special cases of business objects (OE, SA) 2. Study existing conversion techniques 3. Analyze its individual data fields (OE, SA) 4. Determine whether it should be migrated (OE, SA) 5. Create a conversion table (OE, SA) 6. Create a conversion testing plan (OE) 4. Determine the overall conversion strategy (DM, OE, SA) <ol style="list-style-type: none"> 1. Manual or automatic approach 2. Estimate resources required for each approach 3. Compare the approaches 4. Determine the approach </div> <div style="width: 48%;"> <ol style="list-style-type: none"> 5. Further analyze the archiving strategy <ol style="list-style-type: none"> 1. Analyze the types of business objects to be archived (OE, SA) 2. Determine technical archive solution (OE, DM, SAR) <ol style="list-style-type: none"> 1. Identify alternative archiving solutions 2. Study feasibility of each solution 3. Estimate resources required for each archiving solution 6. Further analyze the dependencies to other systems (SM, ST) <ol style="list-style-type: none"> 1. Identify and analyze the impact on the environment of the system to be retired 2. Inform the stakeholders affected by the retirement 7. Determine dates (BF, SM) <ol style="list-style-type: none"> 1. Date when the business objects should be migrated to the new system 2. Date when the system should be retired 8. Identify and analyze changes to be made in the new system <p>Comment: In order to realize the migration process</p> <ol style="list-style-type: none"> 1. Analyze working routines (OE, SA) 2. Determine impact and consequences of these changes (OE, SAR) 3. Determine and conduct the changes in the new system (DM, OE) 4. Test the changes. </div> </div>	
<p style="text-align: center;"><u>Conversion</u></p> <ol style="list-style-type: none"> 1. Develop the automatic conversion method (D) <ol style="list-style-type: none"> 1. Develop scripts 2. Develop automation processes 2. Test the automatic conversion method (OE, D) 3. Conduct automatic conversion (D) 4. Conduct manual conversion (U) 5. Test the conversion results (U) 	<p style="text-align: center;"><u>Closedown</u></p> <ol style="list-style-type: none"> 1. Close down the system to be retired (SM, ST) 2. Remove the dependencies to the environment (SM, ST)

man-days for the whole manual conversion. The estimates for manual conversion were then compared to the estimates made for the automatic conversion. The decision was made that most of the objects were to be automatically converted.

The archiving strategy was determined in this phase as well. Here, one investigated (1) how the data should be archived, (2) the need for accessing it in the future, and (3) the effects of archiving it. Together with the laws and rules as identified in Activity 2, this information provided feedback for deciding upon the technical archiving solution. The criteria used were technical feasibility and cost.

The strategy chosen was to let all the data stay untouched in Indra and Gliid and just to close the two systems for update. The cost of having these systems in

operation was estimated to be very low. The alternative archiving strategies were to move all the data from Indra and Gliid to LH or to build a completely new archive. Both these alternatives were regarded to be too costly.

As a next step, one analyzed dependencies to the interfacing systems. When doing it, one identified and analyzed how they were affected by the closure of Indra and Gliid. The analysis showed that the closure did not imply any major changes and implications to these systems. The only action that was required was to inform their managers about the forthcoming closure. Finally, one determined the date when the business objects should be migrated to the new system and when the old systems should be retired.

It was suspected that the retirement work would lead to some additional changes to be made in the LH system. To identify these changes, one analyzed current working routines, suggested changes to them and their supporting system (LH), and determined the impact and consequences of their implementation. For all the changes identified, one created change requests and sent them to the team responsible for making changes to the LH system.

An example of a major change to be introduced in LH was the implementation of the report generators that were used in the Indra and Gliid systems. An example of a minor change was the creation of a certain data field. All the major and minor changes were tested in the LH system before the *Conversion* phase started.

4.3 Conversion

The *Conversion* phase started only after all the preparations had been successfully conducted. As a first step, one developed the automatic conversion method including scripts and automation processes. This method was then tested with the purpose of estimating the conversion time and of assuring a problem free conversion. After the tests had been successfully passed, one conducted both the automatic and manual conversion on the day as determined in Activity 7 in the *Preparation* phase. The results were finally tested to verify a successful conversion.

4.4 Closedown

In the *Closedown* phase, one closed down the Indra and Gliid systems and removed their dependencies to the adjacent systems. The closure in our case implied that the users could no longer access information in Indra and Gliid.

5. Evaluation

In this section, we make two evaluations. We first present the results of our fifth research method step during which we evaluated our elicited model within FORSAK. We then evaluate it against the current standard retirement process models [2,3].

5.1. Industrial Evaluation

The retirement process model was presented to the project manager responsible for the retirement project. According to her, our model was realistic and it fully reflected the retirement process as performed within the EXIT project. She has, however, observed three minor deficiencies which she believed were very important for a successful execution of a retirement process. These concerned adding two new activities to the first retirement phase and adding a new role.

The first new activity dealt with risk management. According to her, controlling risks was an essential activity within the retirement process. Not doing it implies a critical business risk by itself. The second

activity concerned determination of retirement project budget. According to the project leader interviewed, due to the criticality of retirement projects, it is very important to assign substantial resources to the retirement project. Otherwise, one runs the risk that one underestimates the project and thereby fails with its completion.

Regarding the missing role, it concerned the role of a *System Architect*. According to the project leader, this role is indispensable in all the retirement projects. Not only does this role know the system to be retired but also all its architectural flaws and deficiencies that should not be migrated to the new system.

As a response to these deficiencies, we have added the budget and risk management activities and the *System Architect* role to our model. The modifications are marked with the underlined text written in bold letters in Table 1.

5.2 Evaluation against Current Standards

In this section, we compare our retirement process model with the standard process models as described in [2, 3]. When doing this, we follow the comparison criteria listed in Table 1. Except for the criteria concerning the roles, all the comparison results are outlined in Table 3.

None of the standard process models suggests any roles to be directly involved in the retirement process. Regarding the ISO/IEC standard [3], it only briefly mentions that personnel be trained in retirement actions. The IEEE model [2], on the other hand, mentions a user role, who should be notified about the closure of the system. Within the EXIT project, we have however identified nine different roles. These are listed and described in Section 4.

The broad portfolio of the roles identified in the EXIT project indicates that the retirement project involves the majority of the organizational roles ranging from user to various analyst and design roles, to managerial roles and even to support roles. This, in turn, indicates how complex and comprehensive the retirement process model is.

As illustrated in Table 3, none of the standard process models includes the activities during which one analyzes the retiring and replacing system. We believe that these

Table 3. Our comparison results

Activities	IEEE	ISO/IEC	EXIT
- System Analysis	–	–	+
- Archiving Strategy	–	+	+
- Migration Strategy	–	–	+
-Management of the adjacent systems	–	+	+
- Retirement planning	+	+	+
- Risk management	–	–	+
- Conduct archival	+	+	+

are one of the most important activities within the retirement process. They could be compared to the requirements specification activities. It is a common knowledge that non-recognition of the requirements, irrespective of what type of a project it concerns, does not lead to successful project results. For this reason, we claim that lack of analysis activities is a series deficiency in the standard process models studied.

Only the ISO/IEC 15288 standard [4] suggests the identification of archiving strategies. None of the standard models proposes migration strategies. In our opinion, identification of both these strategies is very important. Identification of the retirement strategy is a must. However, the identification of the migration strategy should be an option. This is due to the fact that not all retiring systems undergo migration. We believe, however, that the inclusion of this strategy in the retirement process model indicates that the retirement process does not exist in a vacuum. Many times, parts of the retiring systems have to be migrated to other new replacing systems or other new archiving systems.

Only the ISO/IEC 15288 standard [4] briefly mentions that the interfaces to the adjacent systems should be considered. None of the standard models suggests how the interfacing systems and their users should be handled. In our opinion, this is a serious omission. Improper management of the adjacent systems may lead to big inconsistencies and problems in their future operation. Hence, we suggest that the interfacing systems and their handling should be highly prioritized in a retirement process.

Both the standard process models studied included the planning activities. However, they only recognized the need for planning. They have not provided any suggestions specific to the retirement planning process.

None of the standard process models studied included risk management. We did not include it either in our preliminary process model outline. Even if a risk management is a separate process, we strongly believe that it definitely should be integrated with the retirement process. Retirement and replacement imply many serious business risks. Not considering them may jeopardize the whole retirement process, and thereby the organization's future business opportunities.

Finally, all the standard process models included the archival activity. This activity however was only briefly mentioned, even in our process model. We suspect that this activity is quite complex. Hence, it should be further studied and explored.

6. Final Remarks

In this paper, we have elicited a retirement process model. Our goal was to provide a basis for creating theory on the domain of a retirement process, to evaluate it against current process standards and provide feedback for their extension. The elicitation process was made within one Nordic financial company.

Our results show that our process model is realistic and that it correctly reflects the EXIT retirement project.

Although, its design is based on only one project, it already may provide a basis for comparing it with current retirement standard models and for making suggestions for their improvements and extensions. These improvements and extensions are the following:

- *Extend the retirement process model with the roles involved in the retirement process.* Given the specific characteristics of the retirement process, it is not always obvious who should do what and why. To fully provide support to the organizations, one needs to compliment the retirement process models with the list of roles and their responsibilities.

- *Include analysis of the system to be retired.* Only then you may make sure that you have not gotten rid of important information.

- *Extend the retirement process model with the migration strategy.* This is a way of indicating that a retirement process model is always conducted in a major context.

- *Provide clear instructions for how to manage the adjacent systems.* This concerns both the adjacent systems and its users.

- *Make suggestions for how to plan a retirement process.* This will help the organizations identify the full coverage of retirement and migration activities necessary for shipping successful project results.

- *Include risk management in the retirement project.* It is only in this way; one may become proactive against many serious business risks in this very critical activity.

Our next step is to create a generic retirement process model. In [5], we have elicited another instance of a retirement process model. This instance substantially differs from the process model elicited herein. For this reason, we believe that we are going to meet a great challenge when trying to consolidate these two process models. We are however prepared to meet this challenge.

REFERENCES

- [1] V. T. Rajlich and K. H. Bennett (2000), "A staged model for the software life cycle," *Computer* 33(7), 2000.
- [2] S. W. Ambler, M. J. Vizdos, and J. Nalbone, "The enterprise unified process: Extending the rational unified process," Upper Saddle River, N. J.: Prentice Hall PTR, 2005.
- [3] IEEE Standard for Developing Software Life Cycle Processes, IEEE Std 10741991, The Institute of Electrical and Electronics Engineers, Inc. 345 East 47th Street, New York, NY 10017-2394, USA, 1991.
- [4] ISO/IEC 15288, Systems and Software Engineering—System life cycle processes, IEEE Std 15288-2008, 2008.
- [5] M. Kajko-Mattsson, R. Fredriksson, and A. Hauzenberger, "Eliciting a retirement process model: Case Study 2," in *Proceedings, International Conference on Innovation in Software Engineering*, IEEE, 2008.
- [6] B. Laurel, "Design research: Methods and perspectives," the MIT Press, 2003.

A Bioinformatics-Inspired Adaptation to Ukkonen's Edit Distance Calculating Algorithm and Its Applicability Towards Distributed Data Mining

Bruce Johnson

University of Tennessee 1508 Middle Drive Knoxville, TN 1-865-974-3461

Email: bjohnson@cs.utk.edu

Received November 25th, 2008; revised November 29th, 2008; accepted December 1st, 2008.

ABSTRACT

Edit distance measures the similarity between two strings (as the minimum number of change, insert or delete operations that transform one string to the other). An edit sequence s is a sequence of such operations and can be used to represent the string resulting from applying s to a reference string. We present a modification to Ukkonen's edit distance calculating algorithm based upon representing strings by edit sequences. We conclude with a demonstration of how using this representation can improve mitochondrial DNA query throughput performance in a distributed computing environment.

Keywords: Bioinformatics-Inspired Adaptation, Calculating Algorithm, Data Mining

1. Introduction

Let Σ be a finite alphabet and let Σ^* denote the collection of finite strings over Σ . Edit distance is a means of measuring similarity between a target and reference string in Σ^* by computing the minimum number of change, insert, or delete edit operations that transform one string into another. The edit distance is a metric [1] and is a means of measuring the similarity between two strings [2].

Wagner and Fischer presented one of the first algorithms for calculating edit distance [3]. Ukkonen improved upon Wagner and Fischer's algorithm (using potentially less time and space) [4,5]. However, a significant performance bottleneck in Ukkonen's algorithm is calculating the length of a longest common prefix (which we refer to as the *degree of agreement*) between two strings.

Let alphabet $\Sigma_d = \{a, c, g, t\}$. Σ_d can be regarded as representing the molecules adenine, cytosine, guanine and thymine respectively. These molecules are collectively known as nucleotides. When covalently bonded together, these molecules become a polymer called a polynucleotide. Two polynucleotides can produce the well-known double helix shape of DNA. The determination of the order in which the nucleotides are covalently bonded together in a polynucleotide is called *sequencing*. The act of sequencing yields a string since each nucleotide in the given polynucleotide maps to one of the members of Σ_d .

The mitochondria are organelles found throughout eukaryotic cells. They are responsible for the production of adenosine triphosphate (ATP), the primary currency by which a cell's energy needs are trafficked [6]. Mitochondria possess DNA (mtDNA). This mtDNA is ultimately responsible for the production of the proteins which regulate the mitochondrion and produce ATP.

We define an mtDNA *string* to be the string that results from sequencing one of the polynucleotides that comprise mtDNA. Anderson et al. [7] were the first scientists responsible for sequencing a human's mtDNA. The mtDNA string they produced is a standard reference and is now known as the Cambridge Reference Sequence (CRS).

Mitochondrial DNA is the subject of much research by forensic scientists because it has features that aid them in their identification of an individual [8].

- 1) It is widely distributed throughout a given cell
- 2) It is always inherited from a child's mother
- 3) It is conservative, i.e., the edit distance between the CRS and a target mtDNA string is very small in comparison to their lengths.

The first feature means that intact mtDNA can likely be extracted from some piece of human detritus such as hair or fingernails.

The second feature means that it is likely that the mtDNA possessed by maternally related individuals is the same. This feature is particularly advantageous for individuals who seek to determine whether the remains of a body belong to their sibling.

With regard to the third feature, we will show that since mtDNA is conservative, the performance of the longest common prefix calculation for Ukkonen's edit distance calculating algorithm can be improved by representing strings as edit sequences. We will show how this feature can improve mtDNA query throughput performance in a distributed computing environment.

2. Preliminaries

2.1 Definitions

We begin by defining edit operations (to streamline exposition, they may be referred to simply as operations).

A nontrivial change operation has the form of $ac\sigma$ and acts on string $\alpha = \alpha_0 \dots \alpha_l$ (provided $0 \leq a \leq l$) to produce $\beta = \beta_0 \dots \beta_l$ where

$$\beta_i = \begin{cases} \beta_i, & \text{if } i \neq a \\ \sigma, & \text{if } i = a \end{cases}$$

In other words, symbol α_a at (address) a is changed to symbol σ .

An insert operation has the form of $ai\sigma$ and acts on string $\alpha = \alpha_0 \dots \alpha_l$ (provided $0 \leq a \leq l$) to produce $\beta = \beta_0 \dots \beta_{l+1}$ where

$$\beta_i = \begin{cases} \alpha_i, & \text{if } i < a \\ \sigma, & \text{if } i = a \\ \alpha_{i-1}, & \text{if } i > a \end{cases}$$

In other words, symbol σ has been inserted into string α at address a .

A delete operation has the form of ad and acts on string $\alpha = \alpha_0 \dots \alpha_l$ (provided $0 \leq a \leq l$) to produce $\beta = \beta_0 \dots \beta_{l-1}$ where

$$\beta_i = \begin{cases} \alpha_i, & \text{if } i < a \\ \alpha_{i+1}, & \text{if } i \geq a \end{cases}$$

In other words, symbol α_a has been deleted from string α .

A sequence of edit operations is referred to as an edit sequence. The concatenation of edit sequence s with t is denoted $s|t$.

Given edit operation e , the function $\&()$ returns e 's address, (i.e. $\&(ad) = a$), the function $\tau()$ returns e 's type (i.e. $\tau(ai\sigma) = i$) and the function $\delta()$ returns the symbol to be inserted or changed, i.e. $\delta(ac\sigma) = \sigma$.

A change operation e is called trivial (with respect to α) if it acts as the identity function on α (i.e. $e(\alpha) = \alpha$). To indicate that is trivial (when is understood) it may be written as $at\sigma$.

The notation $[expression]$ is defined as

$$[expression] = \begin{cases} 1, & \text{if } expression \text{ is true} \\ 0, & \text{if } expression \text{ is false} \end{cases}$$

Given strings $\alpha, \beta \in \Sigma^*$, an edit sequence s taking α to β (i.e. $s(\alpha) = \beta$) is produced by Wagner and Fischer's algorithm [1]. Their algorithm—which we refer to as *WF*—first proceeds by calculating a $(n+1) \times (m+1)$ distance matrix D as follows (where $|\alpha| = n$ and $|\beta| = m$).

$$D_{i,j} = \begin{cases} i, & \text{if } j = 0 \\ j, & \text{if } i = 0 \\ 1 + \min(D_{i-1,j-1} - [\alpha_{i-1} \neq \beta_{j-1}], D_{i-1,j}, D_{i,j-1}), & \text{otherwise} \end{cases}$$

Next, an edit sequence s (transforming α into β) is obtained by the recursive function S

$$S(\emptyset) = \varepsilon$$

$$S(D) = e | S(D')$$

where \emptyset denotes the empty matrix (0 rows, 0 columns), ε denotes the empty edit sequence, and D' is either the result of removing the last D (if case 1 applied), removing the last column from D if case 2 applied or removing both the last row and last column from D (if case 3 or 4 is applied).

$$e = \begin{cases} md, & \text{if } D_{n,m} = 1 + D_{n-1,m} \quad \text{case 1} \\ (m-1)i \beta_{m-1}, & \text{if } D_{n,m} = 1 + D_{n,m-1} \quad \text{case 2} \\ (m-1)c \beta_{m-1}, & \text{if } D_{n,m} = 1 + D_{n-1,m-1} \quad \text{case 3} \\ (m-1)t \alpha_{n-1}, & \text{otherwise} \quad \text{case 4} \end{cases}$$

Given edit sequence $s = S$ transforming α into β , the function $r(s, \alpha, \beta)$ returns the *reduced* edit sequence s_r (with respect to α). Example: let $s = e_0 e_1 e_2 e_3 e_4 e_5 = 0tg \ 1t \ 2d \ 2ig \ 3t \ 4ct$. Then, $s_r = r(s, \alpha, \beta) = e_0 e_1 e_2 = 2d \ 2ig \ 4ct$. Note that

1) $0' = 2$, $1' = 3$ and $2' = 5$

2) both s and s_r map α to β

3) s_r uses the minimum number of edit operations to transform α to β

Edit sequence $s = S(D)$ has the following properties.

1) No edit sequence mapping α to β is shorter than $r(s, \alpha, \beta)$.

2) Addresses of edit operations found in s are nondecreasing.

3) If e_j is a delete edit operation in s , then $\&(e_j) = \&(e_{j+1})$.

4) If e_j is an insert or change edit operation in s , then e_{j+1} has an address that differs from e_j by one.

2.2 Characteristics of Reduced and Non-reduced Edit Sequences

Given edit sequence s , define $\langle s \rangle$ by

$$\langle s \rangle = \sum_{e \in s} [\tau(e) = i] - \sum_{e \in s} [\tau(e) = d]$$

Given $s_r(\alpha) = \beta$ the length of β can be recovered by

$$|\beta| = |\alpha| + \langle s \rangle$$

Let ρ_t be a subsequence of s consisting of trivial change operations, maximal with respect to containment, such that the addresses of successive members differ by one. Such a subsequence ρ_t is called a *trivial change queue*. Example: $s = 0ta \ 1tc \ 2ia \ 3ct \ 4tt$; $\rho_t = 0ta \ 1tc$.

Let ρ_c be a subsequence of s consisting of nontrivial change operations, maximal with respect to containment, such that the addresses of successive members differ by one. Such a subsequence ρ_c is called a *nontrivial change queue*. Example: $s = 0ca \ 1cc \ 2ia \ 3ct \ 4tt$; $\rho_c = 0ca \ 1cc$.

Let ρ_i be a subsequence of s consisting of insert operations, maximal with respect to containment, such that the addresses of successive members differ by one. Such a subsequence ρ_i is called an *insert queue*. Example: $s = 0ia \ 1ic \ 2ca \ 3ct \ 4tt$; $\rho_i = 0ia \ 1ic$.

Let ρ_d be a subsequence of s consisting of delete operations, maximal with respect to containment, such that the addresses of successive members do not differ. Such a subsequence ρ_d is called a *delete queue*. Example: $s = 0d \ 0d \ 0t \ 1ca$; $\rho_d = 0d \ 0d$.

The length of a change or insert queue $\rho = e_y \dots e_z$ is given by $|\rho| = \&(e_z) - \&(e_y) + 1$.

2.3 Recovering Elements of s Using $s_r = r(s, \alpha, \beta)$

Given $s_r = r(s, \alpha, \beta)$, we can recover the trivial change queues removed from s while producing s_r . We will first

consider how to find the locations and then the symbols associated with trivial change queues.

A trivial change queue ρ_i may be a prefix, a suffix or neither a prefix nor a suffix of s_r . In order to find the addresses of members of ρ_i , there are three cases to consider.

Case 1: Queue $\rho_i = e_k \dots e_l$ is a prefix of s_r :

Queue ρ_i is a prefix of s_r if $\&(e_0) > 0$. Furthermore, $k = \&(e_k) = 0$ and $l = \&(e_l) = \&(e_0) - 1$.

Case 2: Queue $\rho_i = e_k \dots e_l$ is a suffix of s_r :

Queue ρ_i is a suffix of s_r if the last edit operation, $e_{m'}$, in s_r has address $\&(e_{m'}) < n = |\beta| - 1$. Furthermore, $k = m' + 1$, $\&(e_k) = \&(e_{m'}) + [\tau(e_{m'}) \neq d]$, $\&(e_l) = n$ and $l = m' + (\&(e_l) - \&(e_{m'+1}) + 1)$.

Case 3: Queue $\rho_i = e_k \dots e_l$ is neither a prefix nor a suffix of s_r :

Queue ρ_i is neither a prefix nor a suffix of s_r if the consecutive edit operations $e_{j'}$ and $e_{(j+1)'}$ in s_r have addresses $\&(e_{j'}) < \&(e_{(j+1)'}) - [\tau(e_{j'}) = d]$. Furthermore, $k = j' + 1$ and $l = (j+1)' - 1$ where $\&(e_k) = \&(e_{j'}) + [\tau(e_{j'}) = d]$ and $\&(e_l) = \&(e_{(j+1)'}) - 1$.

Now that we know how to find the addresses of members of trivial change queues, we need to find their symbols. Given $s_r = r(s, \alpha, \beta)$. Let cell $D_{i,j}$ have a column whose address is that of a trivial change operation. Let function $ni(j)$ return the number of insert edit operations in s_r whose addresses are less than j . Let function $nd(j)$ return the number of delete edit operations in s_r whose addresses are less than or equal to j . In order to find the symbols in trivial change queues, we discovered that $nd(j) - ni(j) = i - j$.

Since $nd(j) - ni(j) = i - j$ it follows that $\alpha_i = \alpha_{j+nd(j)-ni(j)}$. If $e = at \alpha_i$ then we can say that $e = at \alpha_{j+nd(j)-ni(j)}$. Since the address of e is equal to the column j labeled by $D_{i,j}$, we can say that $e = jt \alpha_{j+nd(j)-ni(j)}$. Hence, given α and s_r , we can acquire the address and symbol associated with each trivial change operation in s .

Given element β_x , let $t_r = \text{Partition}(s_r, x)$ return edit sequence t_r whose elements are comprised of those elements of s_r whose addresses are greater than or equal to x . Let $e = \text{GetOp}(s_r, y)$ return the first edit operation found in s_r whose address is greater than or equal to y . Let ρ_i be a trivial change queue, the following pseudocode $\rho_i = \text{Recover}(s_r, x)$ shows the procedure for finding trivial change queues in s_r . The code is initialized by a call to $\text{Partition}(s_r, x)$.

$\rho_i = \text{Recover}(t_r, x)$

1. $e = \text{GetOp}(t_r, x)$

2. if $(e == e_0 \ \&\& \ \&(e_0) > 0)$ //Case 1

2.1. $k = 0$

2.2. $l = \&(e_l)$

2.3. return $(\rho_i = e_k \dots e_l)$

3. if $(e == e_{m'} \ \&\& \ \&(e_{m'}) < n = |\beta| - 1)$ //Case 2

3.1. $k = m' + 1$

3.2. $l = m' + (\&(e_l) - \&(e_{m'+1}) + 1)$

3.3. return $(\rho_i = e_k \dots e_l)$

4. if $(e == e_{j'} \ \&\& \ \&(e_{j'}) < \&(e_{(j+1)'}) - [\tau(e_{j'}) = d])$ //Case 3

4.1. $k = j' + 1$

4.2. $l = (j+1)' - 1$

4.3. return $(\rho_i = e_k \dots e_l)$

5. return \emptyset

3. Calculating the Degree of Agreement Using Edit Sequences

3.1 Motivation for Using Reduced Edit Sequences

At this point, it is productive to ask why we care about reduced edit sequences. Let reference string α be the CRS, target strings β and γ be mtDNA strings and let $s_{r1} = r(s_1, \alpha, \beta)$ and $s_{r2} = s(s_2, \alpha, \gamma)$. Edit sequences s_{r1} and s_{r2} (and reference string α) can be used as a means of representing β and γ respectively. This is significant because large, conservative target strings are represented by edit sequences that are substantially smaller. Hence, calculating the edit distance between β and γ by using α , s_{r1} and s_{r2} , may lead to a more efficient utilization of distributed computing resources for calculating edit distance by increasing network throughput. Furthermore, using α , s_{r1} and s_{r2} can afford forensic experts seeking to find a match for an mtDNA string the ability to store and carry large numbers of mtDNA sequences.

3.2 Our Algorithm

Let β and γ be target strings of lengths m and n , respectively. Let $s_{r1} = r(s_1, \alpha, \beta)$ and $s_{r2} = s(s_2, \alpha, \gamma)$ and let $(0 \leq x_1 \leq m-1)$ and $(0 \leq x_2 \leq n-1)$. We want to know the length of the longest common prefix of the substrings $\beta_{x_1} \dots \beta_{x_1-1}$ and $\gamma_{x_2} \dots \gamma_{x_2-1}$ (i.e. the degree of agreement between β and γ). We will now consider how the degree of agreement between β and γ can be calculated using reduced edit sequences that represent β and γ by examining how our algorithm deals with the different types of edit operations that comprise our edit sequences used to represent our strings.

Case 1: x_1 or x_2 is the address of a member of a delete queue.

In this case, we do not have any symbols to compare; hence, we will simply traverse to the end of the respective queues.

Case 2: x_1 and x_2 are the addresses of members of trivial change queues ρ_1 and ρ_2 , respectively.

Let l_1 be the last member of ρ_1 and let l_2 be the last member of ρ_2 . Let e_1 be a member of ρ_1 and let e_2 be a member of ρ_2 where $e_1 = x_1 c \alpha_w$, $e_2 = x_2 c \alpha_y$, $w = x_1 + nd_1(x_1) - ni_1(x_1)$ and $y = x_2 + nd_2(x_2) - ni_2(x_2)$. If $w = y$, then $\delta((x_1 + n) \alpha_{w+n}) = \delta((x_2 + n) \alpha_{y+n})$ for $0 \leq n \leq \min(g, h)$, where $g = |\{e_1 \dots l_1\}|$ and $h = |\{e_2 \dots l_2\}|$. Hence, the degree of agreement will be $\min(g, h)$.

Case 3: x_1 and x_2 are the addresses of members of ρ_1 and ρ_2 , respectively and neither ρ_1 nor ρ_2 are trivial change queues nor delete queues.

Let e_j and e_k be members of ρ_1 and ρ_2 respectively, and let $\&(e_j) = x_1$ and $\&(e_k) = x_2$. Let e_y and e_z be the last members of queues ρ_1 and ρ_2 , respectively. Let r be the degree of agreement between β and γ . We compare the symbols associated with these queues sequentially using the following loop.

1. $r = 0$

2. $c = 0$
3. $g = \&(e_y) - x_1$
4. $h = \&(e_z) - x_2$
5. while($c < \min(g, h) \&\& \delta(e_{j+c}) == \delta(e_{k+c})$)
 - 5.1. $c = c + 1$
 - 5.2. $r = r + 1$

We now present the pseudocode for the algorithm responsible for calculating the degree of agreement between β and γ using edit sequences.

```

int GetAgreement( $s_{r1}, s_{r2}, x_1, x_2$ )
1.  $t_{r1} = \text{Partition}(s_{r1}, x_1)$ 
2.  $t_{r2} = \text{Partition}(s_{r2}, x_2)$ 
3.  $r = 0$ 
4. for( $i = j = 0; x_1 < |\beta| \&\& x_2 < |\gamma|$ )
4.1.  $i = i + [\tau(t_{r1}[i]) == d]$  //Case 1
4.2.  $j = j + [\tau(t_{r2}[j]) == d]$  //Case 1
4.3.  $u = x_1 + nd(x_1) - ni(x_1)$ 
4.4.  $w = x_2 + nd(x_2) - ni(x_2)$ 
4.5.  $c = 0$ 
4.6.  $\rho_1 = \text{Recover}(t_{r1}, x_1)$ 
4.7.  $\rho_2 = \text{Recover}(t_{r2}, x_2)$ 
4.8. if( $(\rho_1 \neq \emptyset \&\& \rho_2 \neq \emptyset) \&\& u == w$ ) //Case 2
4.8.1.  $g = \{e_1 \dots l_1\}$ 
4.8.2.  $h = \{e_2 \dots l_2\}$ 
4.8.3.  $b = \min(g, h)$ 
4.8.4.  $x_1 = x_1 + b$ 
4.8.5.  $x_2 = x_2 + b$ 
4.8.6.  $r = r + b$ 
4.9. else //Case 3
4.9.1.  $g = \&(e_y) - x_1$ 
4.9.2.  $h = \&(e_z) - x_2$ 
4.9.3. while( $c < \min(g, h) \&\& \delta(e_{j+c}) == \delta(e_{k+c})$ )
4.9.3.1.  $c = c + 1$ 
4.9.3.2.  $r = r + 1$ 
4.9.3.3.  $x_1 = x_1 + 1$ 
4.9.3.4.  $x_2 = x_2 + 1$ 
4.9.3.5. if ( $\delta(e_{j+c}) \neq \delta(e_{k+c})$ )
4.9.3.5.1. return  $r$ 
4.10.  $i = i + c$ 
4.11.  $j = j + c$ 
5. return  $r$ 
    
```

4. Performance Measurements

In this section, we use a lazy implementation of Ukkonen's edit distance calculating algorithm that has as input:

- 1) Ordinary, uncompressed strings
- 2) Strings whose elements are represented as bits
- 3) Strings whose elements are represented using reduced edit sequences

The algorithms responsible for calculating degree of agreement using these strings as input are designated *lo*, *lbp* and *les*, respectively. Note that *les* incorporates the GetAgreement algorithm mentioned above. Furthermore, note that when we speak of performance of the *lo*, *lbp* or *les* algorithms in our measurements, we are in fact referring to either the performance of the *lo*, *lbp* or *les*-invoking version of Ukkonen's edit distance calculating algorithm mentioned above.

4.1 Performance Comparisons between the *lo*, *lbp* and *les* Algorithms

What follows are measurements of the time and memory usage performance of the *lo*, *lbp* and *les* algorithms. The algorithms use as input 500 randomly selected members from a sample of 200,000 randomly generated mtDNA strings. The algorithms were executed on a 700-Mhz Intel Pentium 3 computer using the Redhat 7.0 operating system.

The figures below compare *lo* with *les*, and *lbp* with *les*, respectively. They indicate that, as expected, when the edit distance is small (meaning that the edit sequence used to represent a string is small), the *les* algorithm will finish execution more quickly.

The following tables indicate the time and memory consumed in the execution of our *lo*, *lbp* and *les* algorithms. While the execution time for *les* is beaten by *lbp*, *les* asserts its usefulness by requiring far less memory than *lbp*.

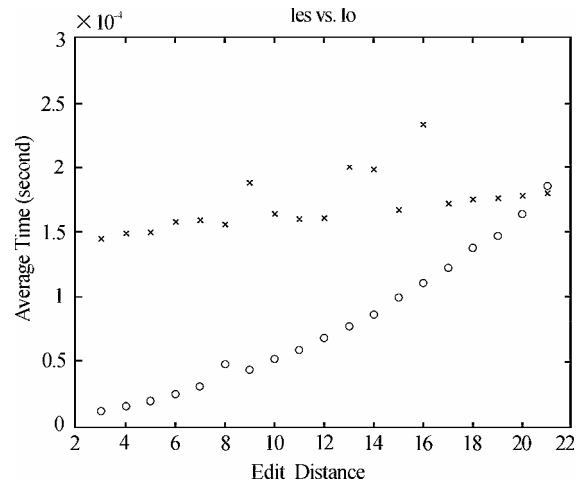


Figure 1. Time used to calculate edit distance using *les* (o) and *lo* (x)

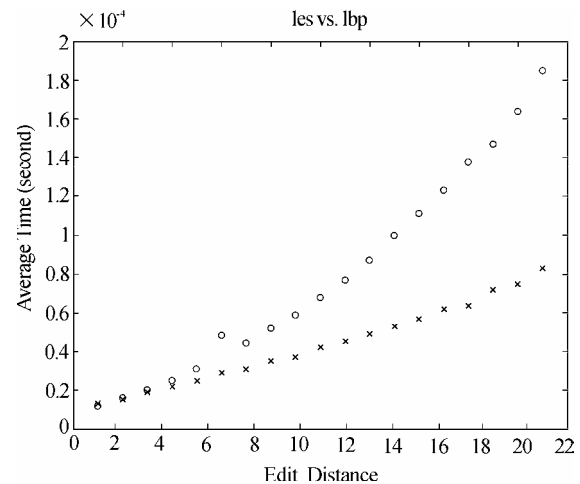


Figure 2. Time used to calculate edit distance using *les* (o) and *lbp* (x)

4.2 Query throughput Performance Comparisons in a Distributed Computing Environment Using the *lo*, *lbp* and *les* Algorithms

A query is defined as an mtDNA string submitted by a client to a server. Query satisfaction is defined as the determination of which mtDNA strings residing on a server fall within an edit distance threshold of the query. Query throughput is defined as the number of edit distance calculations performed in a second while satisfying a query. The following tables provide performance measurements in terms of query strings submitted per second and queries satisfied per second for the *lo*, *lbp* and *les* algorithms in a LAN and WAN distributed computing environment. The algorithms used as input 200,000 randomly generated mtDNA strings. The queries were transmitted on a 1GB LAN where each network node was a 3.2-Ghz Intel Pentium 4 computer using the Debian GNU/Linux 3.1 operating system. The queries were also transmitted on a 54MB wireless WAN where the client and server were 2.2-Ghz and 2.4-Ghz

Table 1. Time consumption (microseconds)

	<i>les</i>	<i>lbp</i>	<i>lo</i>
Average	79	43	172
Minimum	12	13	145
Maximum	185	83	234

Table 2. Memory consumption (bytes)

	<i>les</i>	<i>lbp</i>	<i>lo</i>
Average	337.6	8494	33777
Minimum	300	8494	33777
Maximum	372	8494	33777

Table 3. LAN throughput performance (strings submitted/second)

	<i>les</i>	<i>lbp</i>	<i>lo</i>
Average	3.3e4	1.2e3	310
Minimum	2.9e4	1.1e3	295
Maximum	3.5e4	1.3e3	326

Table 4. LAN query throughput performance

	<i>les</i>	<i>lbp</i>	<i>lo</i>
Average	1.7e4	1.2e3	310
Minimum	5.8e3	1.1e3	295
Maximum	3.4e4	1.3e3	326

Table 5. WAN throughput performance (strings submitted/second)

	<i>les</i>	<i>lbp</i>	<i>lo</i>
Average	9.1e3	353	88
Minimum	7.8e3	340	84
Maximum	9.6e3	362	92

Table 6. WAN query throughput performance

	<i>les</i>	<i>lbp</i>	<i>lo</i>
Average	9.1e3	353	88
Minimum	7.8e3	340	84
Maximum	9.6e3	362	92

Intel Pentium 4 computers, respectively, and were each using the Windows XP operating system. Network performance was measured using Jperf 2.0 [9].

We see that when queries are submitted in a distributed computing environment, the *les* algorithm can accept more query strings transmitted and therefore allows our *les* algorithm to achieve greater query throughput than either the *lbp* or *lo* algorithms.

5. Conclusions

This decade has witnessed three major disasters—the 9/11 attacks, the Indian Tsunami and hurricane Katrina. In the wake of such disasters, identifying people who have perished is of paramount importance.

The usefulness of the *les* algorithm is asserted by the fact that it consumes far less memory than competing algorithms *lo* and *lbp*. This means that greater information throughput may be achieved on a network and thus greater use of distributed computational resources is facilitated.

Moreover, this means that forensic experts can store far more mtDNA sequences using the *les* algorithm than they could if they were using the mtDNA strings required by *lo* or *lbp* algorithms. Having the ability to store a huge number of mtDNA sequences by forensic experts could prove to be a boon by those forensic experts charged with the duty of identifying the remains of people after a major disaster. Having the ability to draw from a vast database of mtDNA strings increases the likelihood that a match can be made between the mtDNA collected and the mtDNA stored in a database.

6. Acknowledgements

We would like to thank Dr. Michael Vose for his kind mentorship and guidance.

REFERENCES

- [1] M. D. Vose, "A formal analysis of edit distance," UT CS Technical Report ut-cs-04-517, February 2004.
- [2] R. O. Duda and P. E. Hart, Pattern Classification (2nd ed.), Wiley Interscience, 2000.
- [3] A. Wagner and M. I. Fischer, "The string-to-string correction problem," Journal of the ACM, 21(1) (Jan. 1974), pp. 168–173, 1974.
- [4] E. Ukkonen, "Algorithms for approximate string matching," International Control 64, pp. 100–118, 1985.
- [5] E. Ukkonen, "On approximate string matching," International Conference Fundamentals of Computation Theory, Lecture Notes in Computer Science, pp. 158:487–495, 1983.
- [6] N. Campbell and J. Reese, Biology (6th ed.), Addison Wesley, 1997.
- [7] S. Anderson, et al., "Sequence and organization of the human mitochondrial genome," Nature, 290(5806) (April 9, 1981), pp. 457–265, 1981.
- [8] K. L. Monson, et al., "The mtDNA population database: An integrated software and database resource for forensic comparison," Forensic Science Communications, 4(2), April 2002. DOI=<http://www.fbi.gov/hq/lab/fsc/backissu/april2002/miller1.htm>.
- [9] <http://iperf.sourceforge.net>.

Designing and Verifying Communication Protocols Using Model Driven Architecture and Spin Model Checker

Prabhu Shankar Kaliappan, Hartmut Koenig

Chair Computer Networks and Communication Systems, Brandenburg Technical University, Cottbus, Germany
Email: {psk, koenig}@informatik.tu-cottbus.de

Received November 21st, 2008; revised November 26th, 2008; accepted November 29th, 2008.

ABSTRACT

The need of communication protocols in today's environment increases as much as the network explores. Many new kinds of protocols, e.g. for information sharing, security, etc., are being developed day-to-day which often leads to rapid, premature developments. Many protocols have not scaled to satisfy important properties like deadlock and livelock freedom, since MDA focuses on the rapid development rather than on the quality of the developed models. In order to fix the above, we introduce a 2-Phase strategy based on the UML state machine and sequence diagram. The state machine is converted into PROMELA code as a protocol model and its properties are derived from the sequence diagram as Linear Temporal Logic (LTL) through automation. The PROMELA code is interpreted through the SPIN model checker, which helps to simulate the behavior of protocol. Later the automated LTL properties are supplemented to the SPIN for the verification of protocol properties. The results are compared with the developed UML model and SPIN simulated model. Our test results impress the designer to verify the expected results with the system design and to identify the errors which are unnoticed during the design phase.

Keywords: UML Modeling, Communication Protocols, Protocol Verification, SPIN Tool

1. Introduction

Due to the huge complexity of modern software systems, it is required to specify precisely what a software component should do and how it should behave [1]. If the final implementation deviates from the expected behavior, then the use of the developed component may fail. This also applies for the development of communicating protocols as they are merely implemented in the software. Currently, most of the protocols are developed through the natural, informal language because it is easy to understand. Special languages known as formal description Techniques (FDTs) have been developed for an unambiguous specification of the software. FDTs distinguish from programming languages by having a formal semantics. Programming languages, such as Java or C++, have only a formally defined syntax. In order to back-up such languages, the *Unified Modeling Language 2* (UML 2) [2] is a collection of semi-formal standard notations and concepts for modeling the software systems at different stages and views during their development.

The development process is supported by the *Model Driven Architecture* (MDA) concept [3], which is initiated from the Object Management Group (OMG). The UML semantics is described in natural English language which includes semantic variation points that leave some semantics issues deliberately open. This desirable property represents a drawback from the verification point of view. To cope with the above problem we propose a 2-phase strategy (see Figure 1). In the first phase, we model the behavior view by UML

state charts and activity diagrams. Next they are translated as a combination of state charts with the semantics of activity diagrams into PROMELA (*PROcess MEta Language*) [4]. In the second phase, we design the communication view using UML sequence and timing diagrams. The model properties are translated into a temporal logic and imported together with the PROMELA code into the model checker SPIN (*Simple Promela INterpreter*) [5] for verification. Furthermore, we illustrate the importance of UML in developing and SPIN in verifying the communication protocols through our approach.

The paper is organized as follows. In section 2 we give a short overview of related work. Section 3 illustrates the MDA approach applied to the development of communication protocols. Section 4 presents our 2-phase design and verification strategy using a case study as example. Some final remarks and an outlook on future work which concludes the paper.

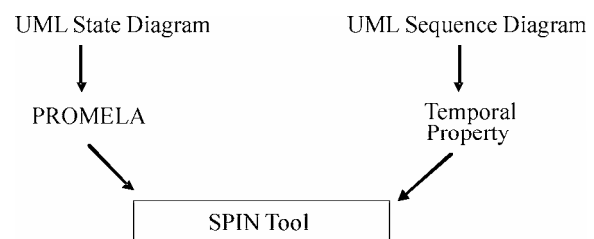


Figure 1. 2-phase strategy

2. Related Works

An approach for the formal verification of UML diagrams, such as class, state and communication diagrams, is presented in [6]. The approach applies an object oriented language, called the *Maude*, for verifying the static and dynamic features of object oriented specifications. Maude is based on rewrite logic. According to [7], there is no proof of correctness (due to the missing UML semantics), when a UML model is translated into PROMELA. To overcome this drawback the static and dynamic verification is carried out individually and integrated into the final validation stage. The verification of the UML class and activity diagrams is illustrated for a simple protocol in [8,9]. The activity diagrams are converted into an FSM (based on behaviors). Thereafter the FSM is converted into PROMELA through an intermediate language. Most of the above specified approaches illustrate how to verify the UML state diagrams. The open issue is how to specify and verify communication protocol properties in detail. According to our concern, the protocols can be efficiently developed if they are verified simultaneously while modeling. In order to fulfill the concern, we specify and verify the protocol properties in the Platform Independent Model (PIM) and the Platform Specific Model (PSM) independently.

3. Architecture Template for Communication Protocols

3.1 Model Driven Architecture

Model driven architecture is an approach to software development based on the modeling and automated mapping of models. MDA has divided its components into two important parts, namely PIM and PSM, which are discussed in detail further as basis.

The *Platform Independent Model* is a model with a high level of abstraction that is independent of any implementation technology [10]. A modeling language capable of generating all the required artifacts such as the Unified Modeling Language is required at this level. According to [3], the PIM provides two basic advantages. First, the person responsible for defining the functionality do not have to take any platform details into the consideration while modeling, which gives the designer a freedom to concentrate and focus only on the logical rule. Second, since the functionality is pure from any implementation details, it is easier to produce implementations on different platforms. The PIM is stored in the Meta Object Facility (MOF) and serves as the input to the mapping step which will produce a *Platform Specific Model*. The PSM's can be described in one of two ways: 1) using UML diagrams (class, sequence, activity etc.) or 2) using interface definitions in a concrete implementation technology (IDL, XML, Java etc), but in both cases the behavior and constraints are

specified using a formal notation (UML diagrams) or an informal notation (natural language). Automated tools will be used to map the platform independent models onto the specific platforms. The final step takes PSM as an input to produce the implementation for a particular platform using a transformation tool.

3.2 Communication Protocols

A communication is carried out between a sender and a receiver over a physical medium using an authorized service provider. The service is provided by means of communicating entities. These entities are active objects exchanging messages with their environment. The service users interact with the entities by exchanging service primitives through *service access points* (SAPs). Each SAP is uniquely mapped to an entity which handles the primitives and maps them on *protocol primitives* or protocol data units (PDUs), respectively, that are sent to the peer entity. The exchange of the protocol primitives is based on rules which are specified by means of a *communication protocol*. A communication protocol describes the interacting behavior of the entities by specifying the timely sequence of the protocol primitives exchanged. Furthermore, the format (syntax) and the meaning (semantics) of the messages are defined.

3.3 MDA and Communication Protocols

The following template for the design of communication protocols consists of three components, namely: the model designer, the model mapper, and the system generator (see Figure 2). These are illustrated with respect to PIM, PSM, and the code generator in the following.

1) Model Designer

The model designer has the task to model the proposed system based on the requirement specification. The modeling is carried out by means of the UML, the *Meta Object Facility* (MOF) for the data repository, and the *Object Constraint Language* (OCL) for the external semantics. The hardware and software may be modeled together or separately. Further on these models are combined by the model integrator (*integrated model*) with the help of external semantics (supplied through OCL), which can be introduced automatically or manually. The advantage of designing hardware and software models independently is that both of them are not considered about the dependency. This gives the developer the freedom to focus on system design rather than on programming details. When considered to the protocol development, the service layer and protocol layer are independently developed in this phase.

2) Model Mapper

The *model mapper* maps the PIM to PSM by means of an appropriate domain specifier. It consists of three different components: the *Domain Specifier* for specifying the target domain, *Transformation Rules*, i.e. a modified Query View Transformation (QVT) [11] is a standard set of rules to map the UML profile to the

particular domain, and (preferably) *UML profiles* for the specification of appropriate models (say protocols). The possible input of the model mapper is UML and the output will be of XML Metadata Interchange (XMI). The transformation process is carried out by an appropriate transformative algorithm which reads the required model (UML profile for communication systems) and applies the QVT rules. The possible outcome of the model mapper is the UML profile based specification models. The transformation method is not strict with the communication system profiles, based on the requirement the profile can be chosen from the repository.

3) Model Checker and Model Verifier

The *model checker* is used to validate the structural behaviors of the developed models. The semantics of PIM are not much validated in this phase because the PIM illustrates only the logical solution to the particular problem. Hence, the structural behaviors are independently verified and combined by the integrated model. The *model verifier* checks the logic after model mapping. In completion of the model mapper phase, the model verifier is introduced to check the static and dynamic behaviors of the mapped model. The verification results from the PIM and PSM are matched by comparing both of the results. Here, the SPIN tool is used along with formal verification techniques to check the behavior of PIM and PSM.

4) System Generator

Finally the code generation is carried after a successful mapping of the model to a particular platform. The target code, such as C++, Java, .Net or SystemC, can be generated by the development tool including the appropriate library files and plug-ins. With help of XMI, which is the (preferable) output code from the previous phase, the code is generated automatically. The generated code is validated thereafter by testing.

By addressing the advantages in the above template, we can consider the top down and bottom up development as

Top down

- Development is from the scratch and to the target code.
- Step by step process, which can be easily debugged or traced.
- Deviation / Refinement are possible at any cost of time.

Bottom up

- Development is from the code and to the specification model.
- Due to the generalized conversion of the XMI, any tool is capable for the conversion of platform independent models.

By the above, the complexity and the development code is systematically reduced with the proposed template.

4. Design and Verification of Communication Protocols

Communication protocols can be distinguished in two different viewpoints: the behavior and the communication oriented one. They can be matched with the UML models as illustrated in the Table 1. The further discussion is based on the above template for protocol development, i.e. we illustrate how the protocol is designed and verified through this template.

4.1 Model Designer

To illustrate the work flow of our method, we use an example case study of the *eXample Data Transfer* (XDT) protocol [12] which is being used as teaching protocol. XDT works on a distributed environment to transfer large files over an unreliable media using the *go back N* principle. The XDT protocol description consists of a service specification and a protocol specification which both include a data format specification. The connection establishment uses a two-way handshake and assumes that the XDT receiver always accepts new connections. The sender makes an initiative for transmission to the receiver by means of an XDAtrequ service primitive. The new connection is indicated by an XDAtind primitive. The protocol indicates the successful connection set up to the sender by XDAtconf.

After this, the data are transferred by means of a DT message. However in certain cases, the service provider may not preserve the order of the data units. In this case, the ABO data unit is initialized to abort the connection.

Table 1. Comparison of protocol and UML viewpoints

Protocol Viewpoints	UML Design Viewpoints
Behavior oriented	Behavior design
What are the behaviors of each communicating entity?	What should happen in the system?
Communication oriented	Interaction design
What is the concrete communication exchange between the entities?	What is the control flow of the data?

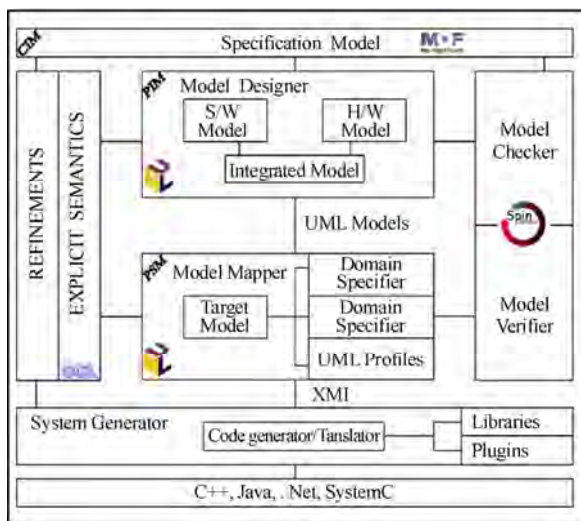


Figure 2. Template for protocol development

This is indicated to the users by a XABORTind service primitive. XBREAKind is initialized to stop the transmission for a certain period, if the *go back N* data buffer is full. The end of transmission is indicated by setting the parameter *eom* in the final data unit of XDATrequ and XDATind primitives. The connection is released implicitly, indicated by an XDISind primitive at the sender and the receiver side, after successfully transmitting the last data unit. The further explanation of the XDT protocol can be found in [12].

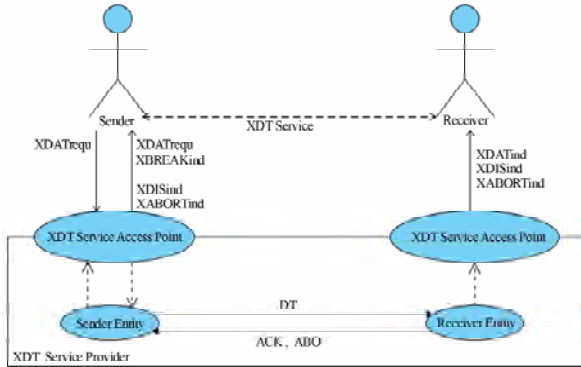


Figure 3. Use case diagram for XDT protocol

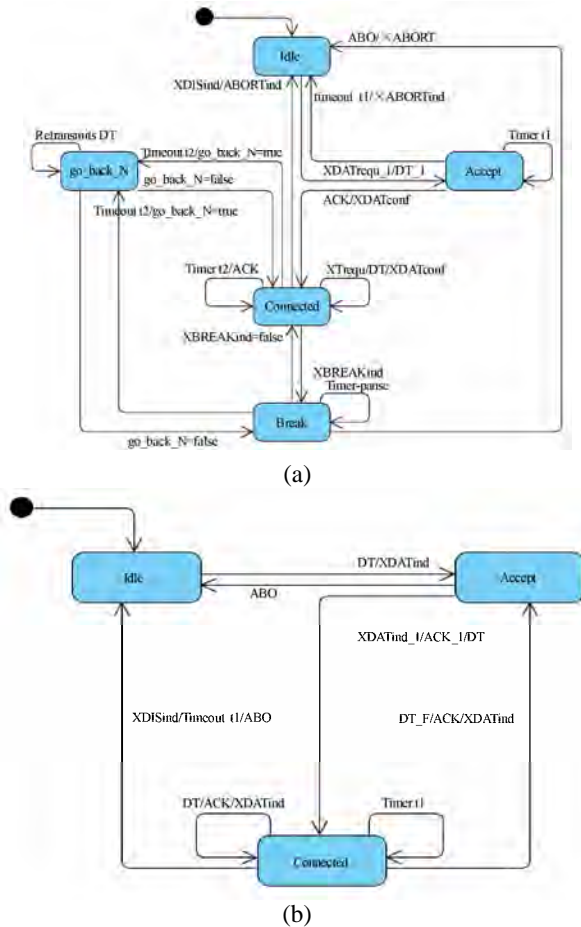


Figure 4. (a) State machine for XDT protocol-sender; (b) State machine for XDT protocol-receiver

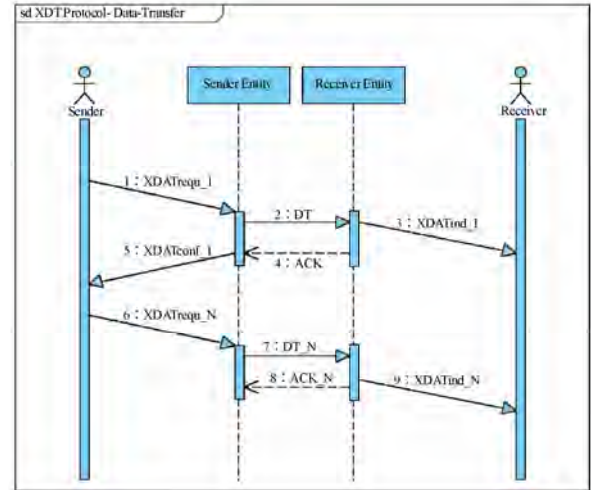


Figure 5. Sequence diagram for XDT data transfer

As a first phase we design the behavior view point by UML use case diagram (see Figure 3) to identify the entities, activity diagrams for the static behaviors, and state machine diagrams (see Figure 4(a), 4(b)) for the dynamic behaviors. Figure 3 i.e. the use case diagram visualize the developer to identify the possible service (XDATrequ, XDATind, XDATconf, XABORTind, XBREAKind, XDISind) and protocol (DT, ACK, ABO) primitives of the protocol. The activity diagrams are used to determine the internal behaviors of the protocol (in which only the semantics are specified). The state machines in Figure 4(a) and 4(b) are the core part for the development. They determine the external behaviors of the protocol by combining the service and protocol primitives. Figure 4(a) and 4(b) represent the sender and receiver part respectively.

As a second phase, we further use the behavior viewpoint as a base and design the communication viewpoint through the sequence (see Figure 5) and timeline diagram to identify the control flow. Figure 5 represents the dynamic behavior of the data transfer state (i.e. connected state in the Figure 4(a) and 4(b)) of the protocol. The same kind of sequence diagram is modeled for all states of the XDT protocol. These sequence diagrams are used further for verifying the protocols.

4.2 Model Checker

To ensure the quality of the developed protocol through the template, the protocol properties (see Table 2) like deadlock, livelock freedom are considered for evaluation. In further we consider our two phase mechanism for verifying these protocol properties.

Phase 1: We retrieve the behavioral viewpoints through the UML use and activity diagrams from the earlier stage. Later these models are translated into the PROMELA via the UML state machine, where the SPIN tool interprets the code. The difference between our approach and others is the following. We use the state machine diagram as a base for the PROMELA translation,

and the semantics from the activity diagrams are added to specify the protocol properties. Since the UML is a semantic-less language, we use the activity diagram as a semantic for the UML state machine model, which is a major advantage. Instead of using external semantics in PIM, the internal semantics makes less complexity and easy usage. The translated PROMELA code is shown in the Figure 6. The protocol entities are described through the keyword *proctype* and the states with *progress*. The code resembles like a C code which is easy to interpret the model. Reference [4] for complete syntax of the PROMELA.

The SPIN model checker executes the PROMELA code and the verification result is produced. The result ensures the quality of the protocol properties like deadlock, livelock, code coverage through its behavior.

Phase 2: To confirm the data flow properties like liveness, the UML sequence diagram is retrieved from the earlier stage and it is converted into a Linear Temporal Logic (LTL) [13]. The LTLs are mathematical annotated formulae to make statements on a linearly progressing time. Since, it is difficult to convert all the UML sequence properties into an LTL; we use another technique known as *Protocol Predictor (PP)*. It identifies the best case criteria in the sequence diagram and marks the event through a unique identifier, e.g. PP:1. The Protocol Predictor is an automated algorithm for UML sequence diagram. It reads the sequence diagram and maintains a periodic log for all service and protocol primitives. The Protocol Predictor has a pre-defined common rules like, the data should be transferred only after a proper acknowledgment; the sequence number should be verified periodically etc. Based on these rules, the algorithm generates the LTL property for the required protocol. In our case, consider that the protocol is working efficiently by transferring the data with sequence number to the receiver. Here we can predict that the sequence number from the sender and receiver should be equal at any time. To do so, we consider the existing LTL property from SPIN as $\square ((p) \Rightarrow (\diamond q))$ with PP:1 and shown in the following code.

PP:1

```
# define p (Data[sequ].sequ == S_N) /* Sender Sequence
number */
# define q (Data[sequ].sequ == R_N) /* Receiver Sequence
number */
/*if p becomes true at one state, q should become true at least
once;
Here by assigning if p (sequence number) is true in Sender,
then q (sequence number) should be true in Receiver */
never { /* !(  $\square ((p) \Rightarrow (\diamond q))$  ) */
Start_S: if
:: (! (q) && (p)) → goto accept_S
:: (1) → goto Start_S ; fi;
accept_S: if :: (! (q)) → goto accept_S; fi; }
```

The idea behind the conversion is that; instead of identifying the worst cases in the communication protocol, we look for the failure of best cases (successful data transmission) which results in identifying the worst cases. This is due to the probability of identifying the worst cases is very less than the probability of best cases. By means of this LTL, it is easy to identify the failure cases like the possibility that sender becomes true and thereafter the receiver remains false forever (or) the possibility that sender becomes *false* before the receiver becomes *true*. Further this code is imported as a supplementary data to the PROMELA code through the SPIN tool for verification. The SPIN model checker validates whether the property holds or not. By investigating this type of combination from the sequence diagram, it is determined that an error-free model is designed. The final result is obtained by transferring five sample protocol primitives from the sender to receiver entity in the SPIN tool. The tool simulates the PROMELA code as a graphical state chart (see Figure 7) to identify the dynamic behaviors and verifies the defined (PP:1) protocol property simultaneously. The verification output from the SPIN tool is shown in Figure 8 with the number of depth reached, state and transition explored. Figure 8 illustrates that no deadlock, livelock is detected in the verification and the five protocol primitives are transferred successfully. The designed model (see Figure 5) is been compared with the SPIN simulated model (see Figure 7). The data transfer phase (second iteration of the Figure 7) is matched perfectly with the designed model. This ensures that the design model is verified for the correctness properties. The advanced LTL property verification represents the model is checked for the protocol properties.

5. Final Remarks

We have discussed about the need of model driven architecture in designing a protocol for dependable systems and the importance of verification. From the above discussion, it is well understood that the combination of MDA technique and the SPIN tool is a reasonable match for the communication protocol development. MDA has the advantage of rapid system development and the SPIN provides a powerful verification mechanism. Since it is an example consideration, the implementation and the

Table 2. Communication protocol properties

Condition	Properties
<i>Absence of Deadlock</i>	The system never enters a state that cannot be left due to a missing or occupied resource
<i>Absence of Livelock</i>	The system never enters cycles that cannot be left due to a missing or occupied resource.
<i>Code Coverage</i>	Each statement defined in the system can potentially be executed.
<i>Liveliness</i>	Each state of the system can be reached from the initial state.
<i>Robustness</i>	The system can react to unexpected, unusual or missing events.
<i>Termination</i>	The final state or an idle state for cyclic systems can always be reached.
<i>Recovery from Failures</i>	The system can recover to a normal state within a limited time after an error has occurred.


```

active proctype Sender_Entity()
/* Sender Protocol Specification */
{
  progress_phase_connect_s:
    XS_XR!Data[1];
  accept_Sender:
    if
      ::XS_XR?Ack[1] -> goto Transfer
    ::else -> goto progress_phase_connect_s;
  fi;
  Transfer:
    atomic {
      progress_phase_Data_Transfer_s:
        If
          ::(!go_back_N) && (!B_break) ->
            sequ = sequ + 1; XS_XR!Data[sequ];
          .....
        fi;
      end_Sender_Entity: }
}

active proctype Receiver_Entity()
/* Receiver Protocol Specification */
{
  progress_connect_r:
    if
      ::XS_XR?Data[1] -> goto
        progress_Data_Transfer_r
      ::else -> goto progress_connect_r;
    fi;
  progress_Data_Transfer_r:
    if
      ::XS_XR?Data[sequ] ->
        .....
      fi;
    end_Receiver_Entity:
  }
}

```

Figure 6. Promela code for XDT protocol

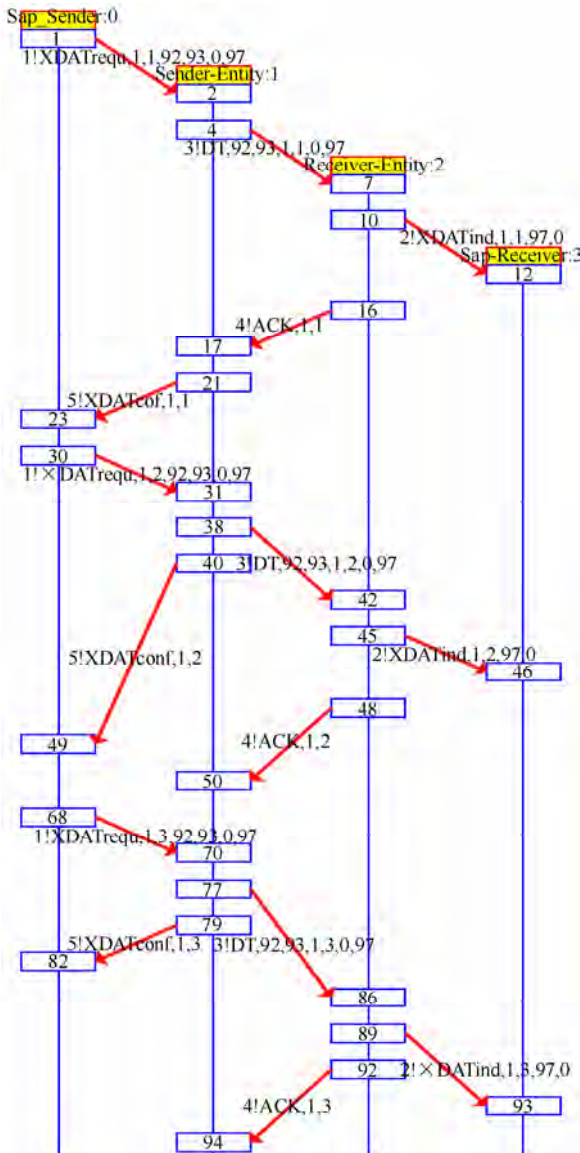


Figure 7. Message sequence chart from SPIN simulation

transformation is carried out manually to test the efficiency of the template. The design and the simulation phase are correlated among each other and the

(Spin Version 5.1.4-27 January 2008)
+Partial Order Reduction

Full statespace search for:

never claim +
assertion violations + (if within scope of claim)
non-progress cycles+(fairness enabled)
invalid end states-(disabled by never claim)

state-vector 692 byte, depth reached 149, errors:0
3816 states, stored (8976 visited)
9673 states, matched
18649 transitions (=visited+matched)
6286 atomic steps
hash conflicts: 2(resolved)

Figure 8. Result obtained from the SPIN tool

effectiveness was measured with the UML sequence diagram and the SPIN chart. As a short term vision, the architecture template and verification strategy has developed on the basis of the MDA approach with the PIM as example implementation.

The further work of the proposed research is to build an automated architecture template for communication protocols. The pitfalls in the existing MDA approach like explicit semantics with standard specifications will be incorporated by proper solutions. It is also planned to develop UML components for the communication protocols. The basic behavior of the protocols will be pre-defined as a component through sequence diagram. Later the sequence diagram will be used in the rapid development as drag-and-drop. Since, we focus to develop a common approach; the same can be used in any protocol development. As a long term vision, the implementation of the developed architecture will be carried out with a real-time peer-to-peer intrusion detection protocol from design to deployment stage.

REFERENCES

- [1] C. Werner, "UML profile for communicating systems," Ph.D thesis, University of Göttingen, Department of Computer Science, 2006.
- [2] Unified Modeling Language, The official homepage of UML, Object Management Group.
<http://www.uml.org>.
- [3] Model Driven Architecture: A Technical Perspective, Architecture Board MDA Drafting Team, Document Number ab/2001-02-04,
<ftp://ftp.omg.org/pub/docs/ab/01-02-04.pdf>, Object Management Group, February 2001.
- [4] Process Meta Language.
<http://www.dai-arc.polito.it/dai-arc/manual/tools/jcat/main/node168.html>.
- [5] G. J. Holzmann, "The model checker SPIN," IEEE Transactions on Software Engineering, 23 (1997) 5: pp. 279-295, 1997.

- [6] M. Farid, G. Patrice, and B. Mourad, "Verifying UML diagrams with model checking: A rewriting logic based approach," Seventh International Conference on Quality Software (QSIC 2007), pp. 356–362, 2007.
- [7] S. Wuwei, C. Kevinon, and H. James, "A toolset for supporting UML static and dynamic model checking," 26th Annual International Computer Software and Applications Conference, 2002.
- [8] B. Prasanta, "Automated translation of UML models of architectures for verification and simulation using SPIN.," 14th IEEE International Conference on Automated Software Engineering (ASE'99), pp. 102–109, 1999.
- [9] S. W. Vitus and J. Padget, "Symbolic Model Checking of UML Statechart Diagrams with an Integrated Approach," 11th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems (ECBS'04), pp. 337–346, 2004.
- [10] A. Kleppe, J. Warmer, and W. Bast, "MDA Explained—The Model Driven Architecture: Practice and Promise," Addison-Wesley, 2003.
- [11] Query, Views, Transformations: A Specification document. [http://www.omg.org/technology/documents/modeling spec catalog.htm](http://www.omg.org/technology/documents/modeling_spec_catalog.htm).
- [12] eXample Data Transfer (XDT) Protocol. http://www.protocol-engineering.tu-cottbus.de/index_xdt.htm
- [13] E. M. Clarke, O. Grumberg, and D. Peled, "Model checking," MIT Press, 1999.

A New Communication Framework for Networked Mobile Games

Chong-wei Xu

Computer Science and Information Systems, Kennesaw State University, USA

Email: cxu@kennesaw.edu

Received November 27th, 2008; revised November 30th, 2008; accepted December 2nd, 2008.

ABSTRACT

This paper introduces a two-layer UDP datagram-based communication framework for developing networked mobile games. The framework consists of a physical layer and a data-link layer with a unified interface as a network communication mechanism. A standalone two-player mobile game, such as a chess game and the like, can be easily plugged on to the communication framework to become a corresponding networked mobile game.

Keywords: Software Framework, Games, Networked Mobile Games, Network Programming, Games in Education

1. Introduction

The game industry is growing very rapidly with a speed of “a near doubling in size in a two-year period” [1]. The mobile devices, especially cell phones, are getting popular and have been a solid part of our daily life. In turn, mobile games are growing even faster than desktop games. According to Informa Telecoms and Media, the worldwide market for mobile games will grow from \$2.41 billion in 2006 to \$7.22 billion by 2011. Juniper Research projects that global revenues of mobile games will grow from \$3 billion in 2006 to \$17.5 billion by 2010 [2].

In addition to the growing and the demand of industry market, technically games including mobile games are the integration of Humanities, Mathematics, Physics, Graphics, Multimedia (images and audios) technologies, Artificial Intelligence, Visualization and Animation, Network Structures and Distributed Computing, programming knowledge and skills, and so on. They provide rich teaching materials and engage students for learning. The demands of the game job market and the special features of gaming itself promote a new pedagogical method by using games for educations [3–7].

We have studied the approach for teaching Object-Oriented Programming (OOP) and Component-Oriented Programming (COP) via gaming [8,9]. Furthermore, we have extended the teaching contents to the field of networked gaming. From the technical point of view, the major difference between the standalone games and networked games is the network communication. Considering the special environment of the networked mobile games, they usually are preferred to be based on the peer-to-peer communication. Thus, the UDP protocol is widely used.

Since a networked game consists of the client site and the server site, which are connected by a communication mechanism, usually the development of a networked game starts the discussion of network programming and applies the client-server model to divide the networked game into two parts. Consequently, the traditional way for developing a networked mobile game is emphasizing on the separation of client and server at the early stage as shown in Figure 1 (a). It is the result that both the client and the server usually are a mixture of the gaming code with the communication code [10–14].

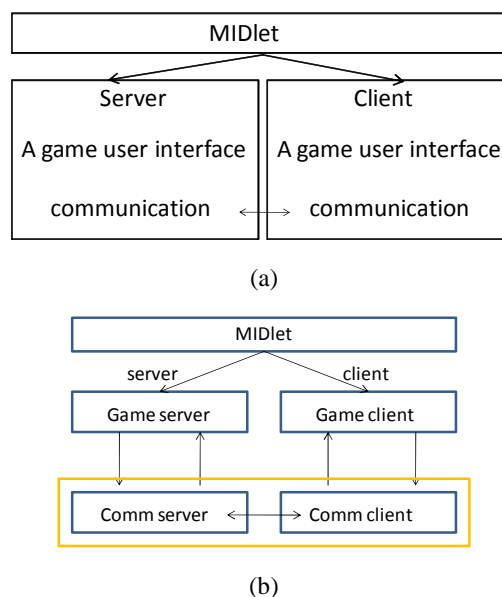


Figure 1. (a) A traditional strategy; (b) A new strategy for developing a networked mobile game

2. A New Problem Solving Strategy

In fact, usually we have had a standalone game already, and then we would like to develop the standalone game to be a networked game. That is, the gaming code and the communication code are the results of two stages of development. Furthermore, if the communication mechanism can be modulated as an independent attachable unit that performs the functionality of passing messages between the client and the server, then it not only increases the reusability and maintainability of the communication code but also makes the transition from the standalone game to the networked game easier.

Following this strategy, we have modulated the communication mechanism as an independent attachable unit with a simple unified interface. Then, two game graphical user interfaces of a standalone game, which represent the client and the server, can be plugged on to the independent communication mechanism through the unified interface for structuring a networked mobile game as depicted in Figure 1 (b). It clearly separates the gaming code from the communication code and allows the communication mechanism can be completely reused for any networked mobile game.

3. Manipulating the UDP Programming Template

For implementing the new strategy, we apply the UDP datagram protocol for making a peer-to-peer environment. By manipulating the UDP datagram communication mechanism in the following steps, the independent attachable communication mechanism has been structured.

First of all, a UDP programming template is derived for depicting its communication mechanism. As we know that J2ME network programming is based on the Generic Communication Framework (GCF) that is illustrated as the connection hierarchy shown in Figure 2. The connection hierarchy has three major interfaces: Content Connection for accessing web data; Datagram Connection for packet-oriented communication; and Stream Connection for stream-based communication. No matter which interface, a foundation class named Connector is used to establish a MIDlet network connection. For mobile games, the more realistic network option is the UDP protocol based on the Datagram Connection because of the limited bandwidth of the mobile phone networks. The programming template of the UDP protocol can be depicted as in Figure 3.

Where, `sdc` stands for server `datagramConnection`; `cdc` stands for client `datagramConnection`. The server builds up a `sdc` and prepares an empty datagram packet `dg` for receiving an input message. And then calls `sdc.receive(dg)`. Whenever the `receive()` method is invoked, the server process is blocked waiting for the incoming message from the client site. When the client builds up its `cdc`, it creates a datagram to contain its out-message and issues `send()` call to send the message out. The server, then, gets the in-message and stores it in the empty datagram packet. This programming template establishes the connection from the client to the server.

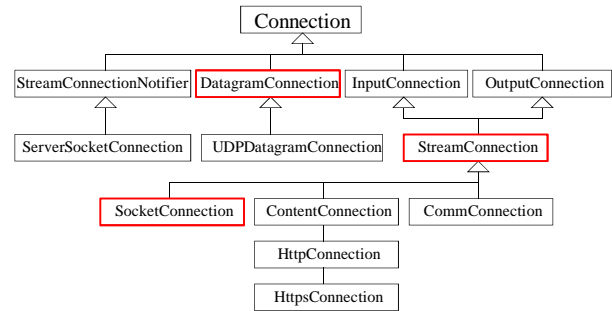


Figure 2. The connection hierarchy of MIDP

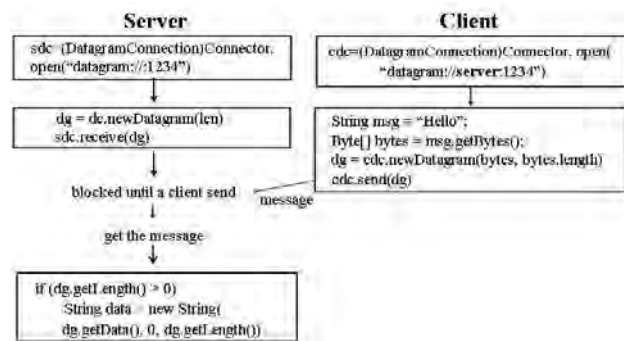


Figure 3. The programming template of UDP protocol

After the server receives the message sent by the client, the server should be able to echo the message back to the client. That is, the client needs to prepare for receiving a message and the server needs to send the message that it just received to the client. The complete programming template is shown as Figure 4. This bi-directional communication mechanism establishes the communication channel and reveals a very symmetric communication system. The only asymmetric codes are referring to the addresses passing, which are marked with the bold face in the figure.

Considering the symmetric scenario, the receiving and sending functions can be moved to a physical layer so that the details of the receiving and sending operations can be hidden. The added physical layer changes Figure 4 to Figure 5.

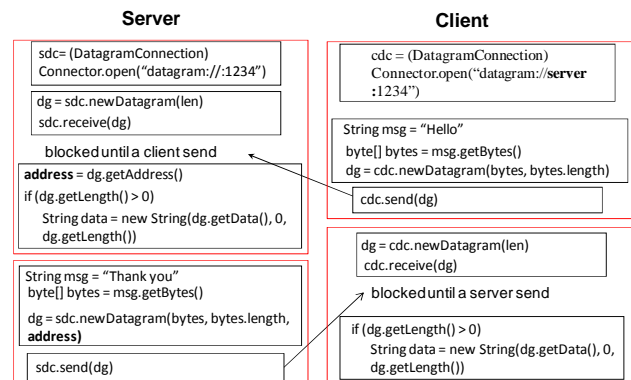


Figure 4. A programming template of the bi-directional communication

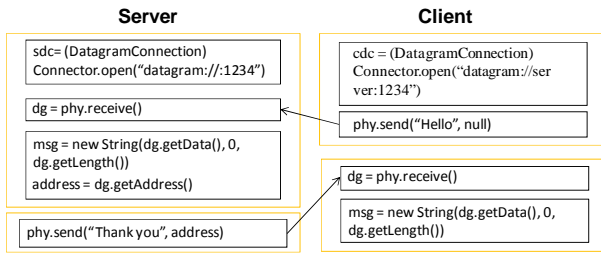


Figure 5. A physical layer for sending and receiving (phy.send() and phy.receive())

Obviously, in order to test the communication mechanism shown in Figure 5, an application should be developed. The simple chat application is selected as an example. Its user interface only needs a TextField component for the user to type in out-messages and a StringItem component for displaying the in-messages. Definitely, the chat communication should be a continuous process until one of partners stops the chatting. For that purpose, a loop is added to keep the chatting process continuous and a sending command is used by the users whenever they make their messages available for sending.

Unfortunately, this version of the chat application experienced both deadlock and duplicate message sending problems. The problems are caused by the structure of the communication mechanism, which uses the physical layer to contain both the phy.receive() and the phy.send() calls. The codes of the phy.receive() and the phy.send() are as follows.

```
public synchronized void send(String msg, String address) {
    byte [] bytes = msg.getBytes();
    try {
        if (address == null) {
            dg = cdc.newDatagram(bytes, bytes.length);
            cdc.send(dg);
        } else {
            dg = sdc.newDatagram(bytes, bytes.length, address);
            sdc.send(dg);
        }
    } catch (IOException ex) {
        ex.printStackTrace();
    }
}

public synchronized Datagram receive(String name)
{
    try {
        if (name.equals("Client")) {
            dg = cdc.newDatagram(100);
            cdc.receive(dg);
        } else if (name.equals("Server")) {
            dg = sdc.newDatagram(100);
            sdc.receive(dg);
        }
    }
}
```

```
}
} catch (IOException ex) {
    ex.printStackTrace();
}
return dg;
}
```

Due to the fact that two methods are shared by both the client and the server, they form critical sections. In order to protect these two critical sections, both methods should be a synchronized method. That is, only one process can enter the methods at a time. Unfortunately, both methods contain sdc (server's datagram connection object) and cdc (client's datagram connection object). As we know that when one process, say the server process, invokes the receive() call, it should be blocked until the other process, the client process, issues a send() call. Therefore, when the server invokes the method phy.receive(), not only the server process itself will be blocked but also the other process, the client process, will be blocked too due to the synchronized protection blocks both resources sdc and cdc inside the phy.receive(). That makes the client process unable to invoke the send() method for sending a message to release the server process since the cdc is blocked. All these together cause a deadlock as depicted in Figure 6.

For overcoming this problem, the synchronized requirement for the phy.receive() has to be released. But, this allows both processes to enter the phy.receive() at the same time and it causes a duplicate message sending.

These two phenomena forced us to move the receive() method out of the physical layer and place it back to the original position and only keep the send() method in the physical layer as Figure 7 shows. This continuous communication mechanism keeps the chat application working. Clearly, it makes both the client and the server consists of three layers: the user interface layer on the top, the physical layer on the bottom, and a layer in the middle, which we gave a name to it as "data link layer".

Based on this layered structure, the user interface layer could be replaced by any game graphical user interface. However, the send Command designed for the chat application cannot be used for games since the players of a game should be able to use key presses for playing the game. Thus, between the user interface layer and the data link layer, a unified interface that consists of two methods: userinterface.receiveMessage(String inMsg)

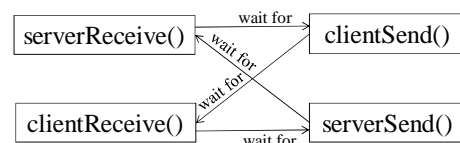


Figure 6. The deadlock scenario

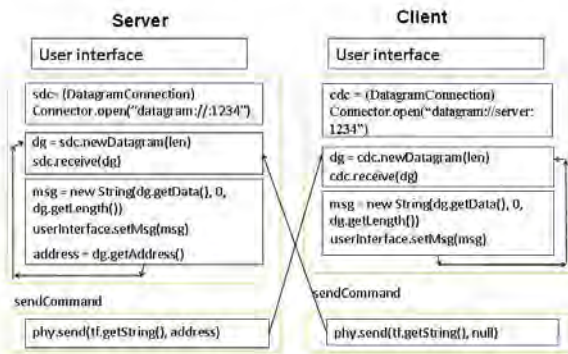


Figure 7. A continuous communication mechanism

and datalink.sendMessage (String outMsg) is inserted. This unified interface plays a role of bridge between the user interface layer and the data link layer. When a player of a game triggers an action that causes the change of the states of the game at one site, the new states will be sent to the other site. The new states carried by the inMsg will be further interpreted by an overloaded method setParameters(inMsg) in the game user interface for controlling the scene of the game. Through this unified interface, the graphical user interface of any standalone game can be easily plugged onto the communication framework as summarized in Figure 8.

4. A Networked Mobile Game Connect 4

We take the Connect4 networked mobile game as an example to demonstrate the application of the framework. This networked game has been described in [10] and implemented according to the traditional method. We have re-designed and re-implemented it by using the new framework. The same game implemented in different strategies enables us to compare the two different strategies for designing and implementing networked mobile games.

For using the framework, a standalone Connect4 game should be developed first, and then add its game graphical user interface on the top of the data link layer in the framework through the unified interface. Because both the client and the server will display the same game user interface, we only need one game user interface for both the client and the server with their own different names, respectively. The standalone game Connect4 that we have developed is described by the simplified UML diagram shown in Figure 9.

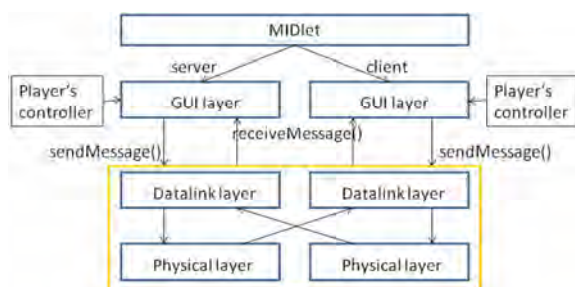


Figure 8. The framework for developing UDP datagram based networked mobile games

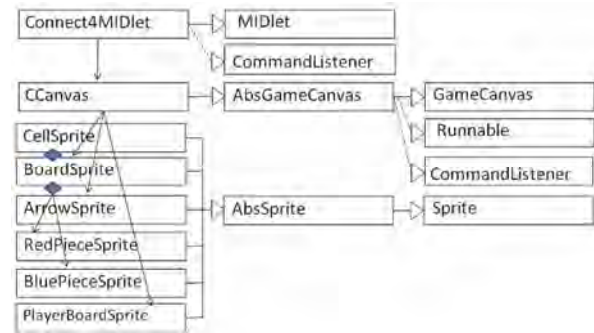
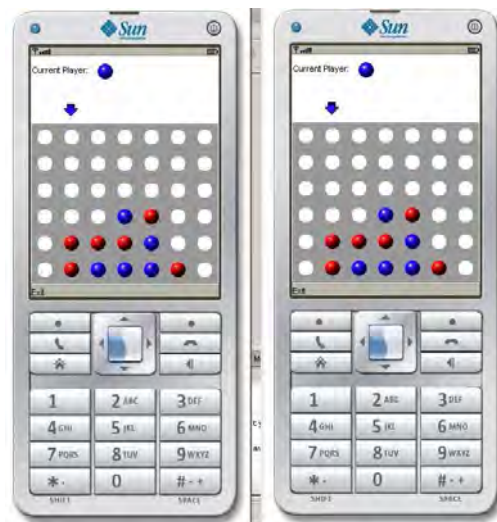


Figure 9. The simplified UML diagram of the standalone game Connect 4

By plugging two game user interfaces with the communication mechanism, the networked mobile game Connect4 is built up as shown in Figure 10 (a).



(a)



(b)

Figure 10. (a) The turn-based networked mobile game Connect4; (b) The event-based networked mobile game Worm. (The left is the server; the right is the client)

The players can control the networked mobile game Connect4 by using the right and the left keys to move the arrow for indicating the target column, and then the players can press the fire key to drop the piece on to the target column. They will take a turn to drop their own pieces with different colors (red and blue). Who will link the four pieces with the same color together either along the horizontal, vertical, or diagonals, who will be the winner of the game.

One of the important design considerations of a networked game is what information should be passed between the client and the server. For the networked mobile game Connect4, there are two kinds of message should be sent. One kind of message only contains a column value, which corresponds to the right or the left key pressing, for synchronizing the arrows' movements in two sites. The second kind of message contains two values: the column number and the current color value, which corresponds to the fire key pressing, for synchronizing the piece dropping. No matter which kind of messages, the user interface layer of the sender site can call the unified interface method `datalink.sendMessage()` to send out a string to the other site. When the receiver site receives the message, its data link layer can use the unified interface method `userinterface.receiveMessage()` to move the received message up to the user interface layer. The user interface layer calls the overloaded method `setParameters()` to interpret the received message for controlling the actions on the receiver site. Due to the fact that both sites have the exact same game logic and under the control of the same parameters, the game user interface layer will display everything the same in both sites, which is the same as the standalone game graphical user interface.

In detail, the networked version needs additional two pieces of code in comparison with the standalone version. One is that the user interface layer needs to instantiate an object of data link layer for sending and receiving messages. The other piece is that when the user interface layer receives messages from the data link layer, it needs to interpret the receiving messages for controlling its own game user interface. In the networked mobile game Connect4, there are two kinds of message are passed so that the user interface layer needs two overloading methods `setParameters()` to interpret the different messages.

5. Conclusions and Future Work

This framework releases the burden for considering a totally different design and implementation between a standalone game with its corresponding networked version. Any take-turn based game can be easily plugged on to the network communication mechanism since it is designed and implemented by following the component-oriented programming philosophy. This structure of the framework allows the data link layer and the physical layer completely reusable. It also makes the standalone

game reusable up to 90% when it will be developed to be a networked game. The game logic wouldn't be touched for both the standalone and the networked versions and all required parameters will be passed along the channel for communication.

Besides supporting networked mobile game development, this framework is also a practical tool for teaching network programming since the developing process of the framework is a manipulation of the UDP protocol. From the manipulation process, students can better understand the functionality of the protocol. It also promotes a sequence of analysis and synthesis processes and enhances students' problem solving ability. Going through the process for developing the framework, we guide students to explore the essential principles of network communication and enrich their foundation on object, module, and component oriented philosophy.

The networked mobile game Connect4 is a turn-based game. Many standalone mobile games played by two competitors, such as a tic-tac-toe, a chess game, an Othello game, and the like belong to this category. These games send and receive messages in a sequential order. The other category of networked games are event-based, where input events made by the players can occur at any time and any player can interact with the game at any time in any order. That is, the messages sent and received are in a concurrent matter. We have developed a networked version of the classic Worm game using the framework, which has two Worms. One player controls one worm for competing to eat the treats as shown in Figure 10 (b). Its functional behaviors need more deeply observations.

This framework is based on the UDP datagram protocol since it supports peer-to-peer model of communications. That limits the number of players to two. What if more players would like to join? Furthermore, the clients of networked mobile games are better to be a thin client since mobile devices have limited supports on their resources. For realizing a thin client, we'd better to move more codes, especially the game logic that is shared by both sites, to be resided on the server site so that two clients don't need to carry them. How to satisfy these requirements? These are the topics that we need to further explore.

6. Acknowledgement

This project was partially supported by the Scholarship of Teaching and Learning Team (STLT) fund, The Center for Excellence in Teaching & Learning (CETL), Kennesaw State University, 2007–2008.

REFERENCES

- [1] M. Zyda, "Educating the next generation of game developers," Computer, IEEE, June 2006.
- [2] F. Chau, "Mobile gaming aims for mass market," 2006. <http://www.smackall.com/viewresource.php?resource=17>.

- [3] M. Mayo, "Games for science and engineering education," CACM, Vol. 50, No. 7, July 2007.
- [4] M. Zyda, "Creating a science of games," CACM, Vol. 50, No. 7, July 2007.
- [5] J. Schollmeyer, "Games get serious," Bulletin of the Atomic Scientists, 2007.
http://www.thebulletin.org/article.php?art_ofn=ja06schollmeyer_100.
- [6] A. Phelps, K. Bierre, and D. Parks, "MUPPETS: Multi-user programming pedagogy for enhancing traditional study," CITC4'03, Lafayette, Indiana, USA, October 16–18, 2003.
- [7] K. Bierre and A. Phelps, "The use of MUPPETS in an introductory java programming course," SIGITE'04, Salt Lake City, Utah, USA, October 28–30, 2004.
- [8] C. W. Xu (2007), "A hybrid gaming framework and its applications," The International Technology, Education and Development Conference 2007 (INTED2007), Valencia, Spain, pp. 30000_0001.pdf, March 7–9, 2007.
- [9] C. W. Xu (2008), "Teaching OOP and COP technologies via gaming," in book "Handbook of research on effective electronic gaming in education," Edited by Richard E. Ferdig, University of Florida, pp. 508–524, IGI Global, 2008.
- [10] M. Morrison, "Beginning mobile phone game programming," Sams, 2005.
- [11] C. Hamer, "J2ME games with MIDP2," Apress, 2004.
- [12] J. Fan, E. Ries and C. Tenitchi, "Black art of java game programming," Waite Group Press, 1996.
- [13] A. Davison, "Killer game programming in java," O'Reilly, 2005.
- [14] D. Brackeen, B. Barker, and L. Vanhelsuwe, "Developing games in Java," New Riders, 2004.

Storing and Searching Metadata for Digital Broadcasting on Set-Top Box Environments

Jong-Hyun Park¹, Ji-Hoon Kang²

¹Software Research Center, Chungnam National University, Gung-Dong, Yuseong-Gu, Daejeon, 305-764, South Korea, ²Dept. of Computer Science and Engineering, Chungnam National University, Gung-Dong, Yuseong-Gu, Daejeon, 305-764, South Korea
Email: {¹jonghyunpark, ²jhkang}@cnu.ac.kr

Received October 30th, 2008; revised November 10th, 2008; accepted November 14th, 2008.

ABSTRACT

Digital broadcasting is a novel paradigm for the next generation broadcasting. Its goal is to provide not only better quality of pictures but also a variety of services that is impossible in traditional airwaves broadcasting. One of the important factors for this new broadcasting environment is the interoperability among broadcasting applications since the environment is distributed. Therefore the broadcasting metadata becomes increasingly important and one of the metadata standards for a digital broadcasting is TV-Anytime metadata. TV-Anytime metadata is defined using XML schema, so its instances are XML data. In order to fulfill interoperability, a standard query language is also required and XQuery is a natural choice. There are some researches for dealing with broadcasting metadata. In our previous study, we have proposed the method for efficiently managing the broadcasting metadata in a service provider. However, the environment of a Set-Top Box for digital broadcasting is limited such as low-cost and low-setting. Therefore there are some considerations to apply general approaches for managing the metadata into the Set-Top Box. This paper proposes a method for efficiently managing the broadcasting metadata based on the Set-Top Box and a prototype of metadata management system for evaluating our method. Our system consists of a storage engine to store the metadata and an XQuery engine to search the stored metadata and uses special index for storing and searching. Our two engines are designed independently with hardware platform therefore these engines can be used in any low-cost applications to manage broadcasting metadata.

Keywords: Digital Broadcasting, Metadata Management, Storing and Searching XML Data, XQuery Processing, TV-Anytime metadata, Set-Top Box

1. Introduction

Digital broadcasting is a novel paradigm for the next generation broadcasting. Its goal is to provide not only better quality of pictures but also a variety of services that is impossible in traditional airwaves broadcasting [1]. One of the important factors for this new broadcasting environment is the interoperability among applications since the environment is distributed. As the digital broadcasting is evolving to more complex and diverse environment due to rapid increase of channels and content, the broadcasting metadata becomes increasingly important. Therefore a standard metadata for digital broadcasting is required and TV-Anytime metadata [2] that is proposed by the TV-Anytime Forum is one of the metadata standards for digital broadcasting [3].

A Set-Top Box, which is called personal digital recorders (PDR), is responsible for receiving and managing the digital content and its metadata. Currently, a Set-Top Box is designed with limited hardware and relatively software. Therefore, it is necessary to develop technologies for effi-

ciently storing of metadata and searching stored metadata based on The Set-Top Box with low-costing and low-setting. Of course, several researches have already proposed some methods for managing metadata on digital broadcasting environment for these necessities [4]. However, we cannot confirm whether their methods run efficiently in a Set-Top box environment because they do not consider characteristics of a Set-Top Box. We have also proposed the method for efficiently managing the broadcasting metadata in a service provider before this study [4]. The result of our research was more effective than other methods. However, to apply our previous methods into Set-Top Box has several problems such as small storage, memory size, and limited software. Consequently, there are some issues to apply general approaches for managing the metadata into Set-Top Box and we have to consider these issues.

In this paper, we propose a method for storing and searching broadcasting metadata. Also we implement the prototype using the proposed method and evaluate our method on a Set-Top Box environment with low-cost and low-setting.

²He is a corresponding author

The remainder of this paper is organized as follows. Section 2 describes the related work. Section 3 and section 4 shows the index for Broadcasting metadata and a method for storing and searching metadata by our prototype system, respectively. In section 5, we describe the conformance evaluation and finally, section 6 provides concluding remarks.

2. Related Work

TV-Anytime forum is organized to develop specifications to enable services based on Local Storage and TV-Anytime Metadata is one of these specifications. TV-Anytime Metadata is used to describe various TV contents and is identified by CRID (Content Reference Identifier). The metadata allows consumer to find, navigate and manage content from a variety of sources, for example, broadcast, TV, internet. XML is the “representation format” used to define the schemas of the TV-Anytime Specification. Also, TV-Anytime metadata is technically defined using a single XML schema, so it is comprised of XML data. Figure 1 shows the structure of TV-Anytime metadata and Figure 2 is its sample instance.

TV-Anytime metadata is technically defined using single XML schema, and it's comprised of XML data. Therefore the method for storing and searching TV-Anytime metadata relates with the method for XML data. Many researchers have investigated different ways of storing XML data in relational databases [4,5,6,7], native XML databases [8,9], and file systems [10,11]. Some researches including our previous research investigated methods for storing the broadcasting metadata into relational database and searching stored metadata [4,5]. [4,5] support both XPath and XQuery languages for searching. So, two systems

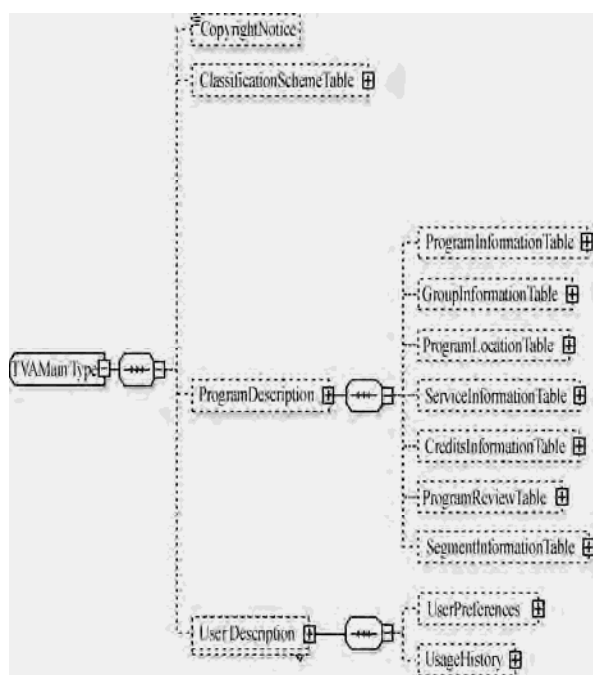


Figure 1. The structure of TV-Anytime Metadata

```
<TVAMain version="1">
  <ProgramDescription>
    <ProgramInformationTable>
      <ProgramInformation
        programId="crid://www.kbs.com/KBSNews9103052300001">
        <BasicDescription>
          <Title type="main">KBS News 9</Title>
          <Synopsis> Bank of Korea Cuts Key Rate, Kim Yu-na Captures
            Skate America Title </Synopsis>
          <Keyword> Main News </Keyword>
          <Keyword> Night News </Keyword>
        </BasicDescription>
      </ProgramInformation>
    </ProgramInformationTable>
    <ProgramLocationTable>
      <OnDemandProgram>
        <Program
          crid="crid://www.kbs.com/KBSNews9103052300001"></Program>
          <ProgramURL>D:\Media\News\news_9.mpg</ProgramURL>
        </OnDemandProgram>
      </ProgramLocationTable>
      <SegmentInformationTable timeUnit="PT1001N30000F">
        <SegmentList>
          <SegmentInformation segmentId="SID_0_0_148">...
        </SegmentInformation>...
      </SegmentList>
    </SegmentInformationTable>
  </ProgramDescription>
</TVAMain>
```

Figure 2. A sample instance of TV-Anytime Metadata

have a module to convert from user query to SQL query and use a specialized indexing method for efficient searching (quick processing of selection, projection, and join). However these two systems use a commercial relational database management system to manage the large volume of metadata because they only focused on service provider systems. Of course, it seems that it is a natural choice to use the RDBMS or Native XML DB because the content service provider has to manage not only the large volume of broadcasting metadata but also a lot of multimedia contents. However, their cost is expensive for STB with low-cost and low-setting.

3. Index for Broadcasting Metadata

In order to store broadcasting metadata, we select the file system because of the cost and hardware power. Although we choose the file system, the basic idea for storing is similar to our previous approach for storing TV-anytime metadata into a relational database. In other words, the basic approach for storing is based on binary approach [12] and the approach for assigning an identifier into a node is the Dewey number labeling [13] to keep a parent-child relationship.

Also we use the path table concept [14] for direct accessing to every nodes and node position concept for obtaining partial document from the metadata instance. Every node which has same name is stored in a single file and information for searching is addressed by the index file. Figure 3 shows the structure for indexing a ‘b’ node.

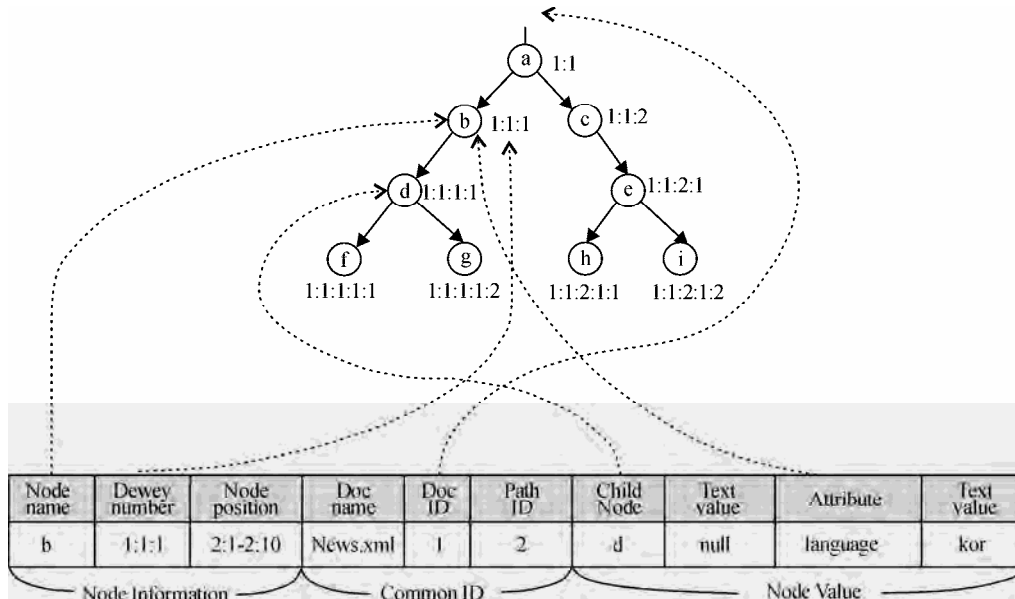


Figure 3. The structure of index

The structure for indexing a node consists of node information part, common ID part, and node values part. In the node information part, we store the name, ID, and position which address the position of current node in original TV-Anytime metadata instance. The common ID part includes the name and ID of TV-Anytime metadata instance and Path ID which links with the XPath expression from root node to current node. The node value parts stores the information of child nodes and attribute nodes.

Figure 4 shows an example XQuery query, Path Index, Node Index and document tree for obtaining result of the query briefly. In order to process the example XQuery query, a node has to satisfy following conditions. The full path expression to 'd' node from root node is 'a/b/c/d', and its value have to contain "KBS News 9". Also the parent node 'c' of the 'd' node must have 'Month' attribute and its value have to equal to 'May'. If a node satisfies these conditions, we can obtain the partial documents of TV-Anytime metadata instance including the node by the Node_Position.

4. Metadata Management System for Storing and Searching

The goal of Metadata Management System is to store and search metadata efficiently in a Set-Top Box environment for digital broadcasting. Figure 5 shows the architecture and function of the metadata management system in the Set-Top Box. Our metadata management system consists of the Storage Engine and the XQuery Engine.

As shown in Figure 6, Storage Engine provides basically four interfaces: InsertDoc, DeleteDoc, UpdateDoc, and GetDoc for inserting, deleting, updating, and retrieving a metadata instance, respectively. In order to generate and store an index file including a metadata, InsertDoc parses the metadata received from Metadata Generator or

Metadata Editor and then extracts and stores the information from the parsing Tree. DeleteDoc deletes the metadata matched with the user-inputted CRID. UpdateDoc deletes the old metadata that has the same CRID as the new metadata, and then inserts the new metadata. Since XQuery doesn't support update of XML data, we use the delete and insert instead of update command.

In this paper, we propose to use XQuery as query language for searching the broadcasting metadata. Since XQuery is standard query language proposed by W3C for querying XML data, it guarantees interoperability between digital broadcasting applications including a Set-Top Box. An XQuery Engine consists of an XQuery parser module for query validation and a SearchDoc module for query execution. The input of XQuery Engine is the XQuery query, and its output is either the whole document or one part of the document. Figure 7 shows the architecture of XQuery Engine for a search of stored metadata.

(1) Input XQuery query

```

For $d in input ("TVAnytime")
For $p1 in $d/a/b/c
For $p2 in $p1/d
Where contains (string($p2), "KBS News9") and $p1/@
Month="May"
return <returns> {$p2} </returns>

```

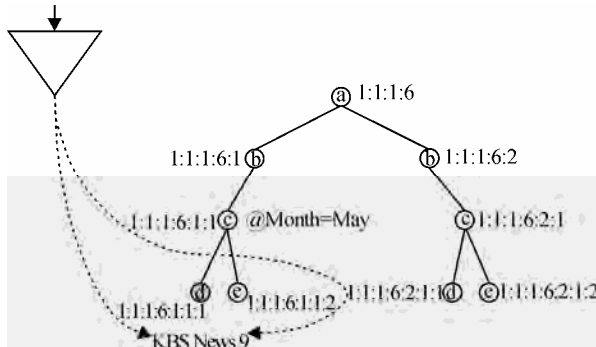
(2) Path Index

Path ID	Full Path expression
62	a/b/c
63	a/b/c/d
64	a/b/c/e

(3) Each Node Index

Node_Name	Dewey_Number	Node_Position	Doc_Name	Doc ID	Path ID	Child_Node	Attribute	Text_Value
d	1:1:1:6:1:1:1	9:13~9:45	News01.xml	1	63	null	null	KBS News 9
c	1:1:1:6:1:1	92:19~97:20	News01.xml	1	62	1:1:1:6:2:2:1	Month	null

(4) Document Tree



(5) Result Composer

<returns>KBS News 9</returns>

Figure 4. An example for processing a XQuery query

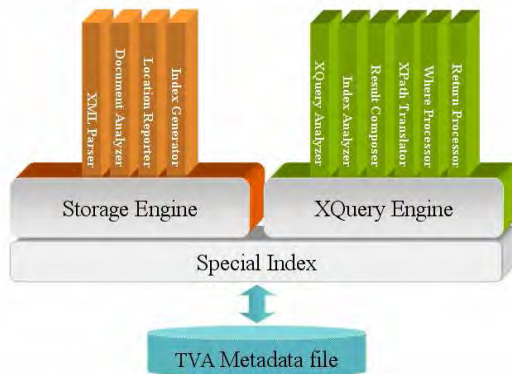


Figure 5. The architecture of TV-anytime metadata management system

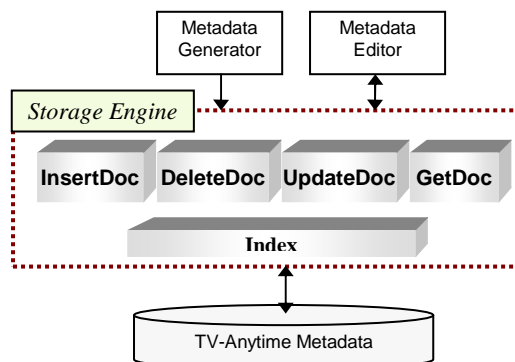


Figure 6. The architecture of storage engine

XQuery Analyzer gets a query in XQuery, parses the query using an XQuery parser and generates its syntax tree. XPath Translator module creates an XPath expres-

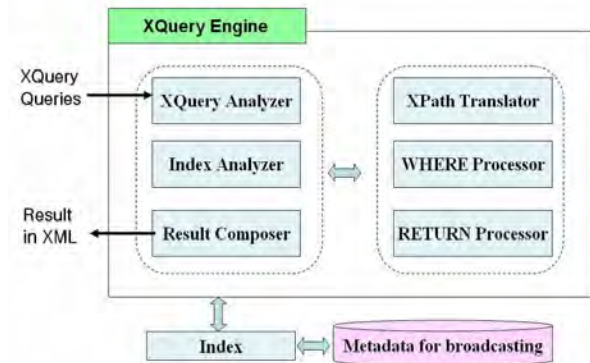


Figure 7. The architecture of XQuery engine

sion which consist of full path expression to current node from root node by merging XPath expressions defined in FOR and LET clauses in XQuery queries. WHERE Processor and RETURN Processor are used for processing conditions defined in a WHERE clause and for constructing the result structure defined in RETURN clause, respectively. Index Analyzer parses the index files and generates the information for obtaining result metadata fragments from the storage by using the selected index. Result Composer constructs the final result using the result structure and result metadata fragments.

5. Performance Evaluation

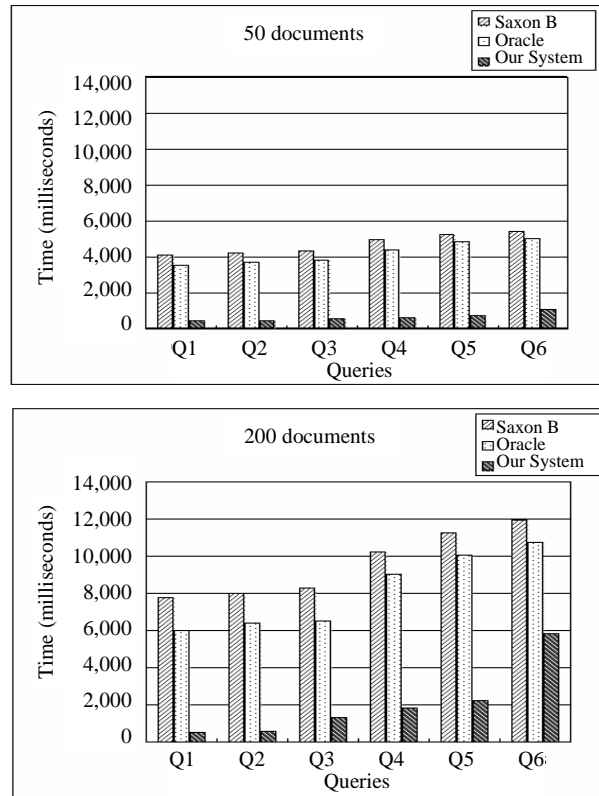
In order to evaluate whether our choice of the strategies for the issues is relevant, we compare our prototype system with other general-purpose XQuery Engine and test their performance for various typical queries. We select two popular general XQuery Engines. One is the Oracle XQuery Engine [10]. The other is a Saxon-B XQuery Engine [11]. Two XQuery Engine it is all free source, a JAVA base, and a head of a family general XQuery Engine. The experimental setup is as follow: the CPU is Intel Pentium III Process 750 MHz, the memory size is 256 MB, the JDK version is 1.4 and the OS is LINUX 2.4.2.

Our system uses XQuery, which is a sub set of XQuery 1.0 (e.g. is not support 'OR' in WHERE clause and '/' in XPath path express). From the previous work [4, 15, 16, 17], we have found that the query processing performance depend on the XPath expression, number of predicates, and result size. By considering these factors, I use the XQuery in Table 1.

We omit some expressions in example queries except Q1. For example, the constructor '<Results>' is omitted because that is the same as in Q1. The queries Q1, Q2 and Q3 use single condition which is declared in the WHERE

Table 1. Example XQuery queries for experiment

	<i>WHERE Conditions / RETURN Value</i>
	<i>XQuery query</i>
	<i>Single condition/ Single terminal node</i>
Q1	<pre> <Results>{ for \$d in input("TVAnytime") return <Result>{ for \$p1 in \$d/TVAMain/ProgramDescription/ ProgramInformationTable/ProgramInformation for \$p2 in \$p1/@programId for \$p3 in \$p1/BasicDescription/Title where \$p2="crid://www.kids17.net/amigonme 103042200049" return <node>{ \$p3 }</node> }</Result> }</Results> </pre>
	<i>Single condition/ Single root node</i>
Q2	<pre> for \$p1 in \$d/TVAMain for \$p2 in \$p1/ProgramDescription/ProgramInformation Table/ProgramInformation/@programId where \$p2="crid://www.kids17.net/amigonme 103042200049" return <node>{ \$p1 }</node> </pre>
	<i>Single condition/ Multiple terminal nodes</i>
Q3	<pre> for \$p1 in \$d/TVAMain for \$p2 in \$p1/ProgramDescription/ProgramInformationTable/ ProgramInformation/BasicDescription/Genre/Name where \$p2="Education" return <node>{ \$p1 }</node> </pre>
	<i>Three conditions/ Single root node</i>
Q4	<pre> for \$p1 in \$d/TVAMain for \$p2 in \$p1/ProgramDescription/ProgramInformation Table/ProgramInformation/BasicDescription for \$p3 in \$p2/Language for \$p4 in \$p2/ProductionDate/TimePoint for \$p5 in \$p2/ReleaseInformation/ReleaseDate/ DayAndYear where \$p3="ko" and \$p4>="2006" and \$p5="2006-04-14" return <node>{ \$p1 }</node> </pre>
	<i>Five conditions/ Multiple root nodes</i>
Q5	<pre> for \$p1 in \$d/TVAMain for \$p2 in \$p1/ProgramDescription/ProgramInformation Table/ProgramInformation/BasicDescription for \$p3 in \$p2/Language for \$p4 in \$p2/ProductionDate/TimePoint for \$p5 in \$p2/ReleaseInformation/ReleaseDate/ DayAndYear for \$p6 in \$p1/ProgramDescription/ProgramLocation Table/BroadcastEvent/Live/@value for \$p7 in \$p1/ProgramDescription/ServiceInformation Table/ServiceInformation/Name where \$p3="ko" and \$p4>="2006" and \$p5="2006-04-14" and \$p6="true" and \$p7="KBS" return <node>{ \$p1 }</node> </pre>
	<i>Three conditions/ Multiple terminal & root nodes</i>
Q6	<pre> for \$p1 in \$d/TVAMain for \$p2 in \$p1/ProgramDescription/ProgramInformation Table/ProgramInformation/BasicDescription for \$p3 in \$p2/Title for \$p4 in \$p2/Language for \$p5 in \$p2/ProductionDate/TimePoint for \$p6 in \$p2/ReleaseInformation/ReleaseDate/ DayAndYear for \$p7 in \$p1/ProgramDescription/ProgramLocation Table/BroadcastEvent/Live/@value for \$p8 in \$p1/ProgramDescription/ServiceInformation Table/ServiceInformation/Owner for \$p9 in \$p1/ProgramDescription/CreditsInformation Table/PersonName for \$p10 in \$p1/ProgramDescription/ServiceInformation Table/ServiceInformation/ParentService where \$p3="KBS News 9" and \$p4="ko" and \$p5>="2006" and \$p7="true" and \$p8="KBS" return <node>{ \$p3, \$p4, \$p5, \$p6, \$p9, \$p10 }</node> </pre>

**Figure 8. Comparison of query processing times**

clause. However, the result data sizes are expected different because the result of each query is a leaf node, an root node, and multiple root nodes together with their descendent nodes, respectively. Q4, Q5 and Q6 use different number of conditions. The return value of each query is a single root node, multiple root nodes, and multiple terminal and root nodes, respectively.

Figure 8 summarizes the performance. The numbers of the test data are 50 and 200 TV-Anytime metadata instances respectively. The result shows that our system outperforms other methods for any queries except Q6. In case of Saxon B and Oracle, the complex queries Q4 and Q5, takes more execution time than simple query Q1, Q2, and Q3. However, our system does not so depend on the queries. In case of our system, Q6 takes more execution time than the other queries since we need time to compose result. However the case of Q6 is not general, because the result size of user queries is not large volume in a Set-Top Box, generally.

Figure 9 summarizes the scalability property of the systems. The size of the test data is 50 documents, 100 documents, 150 documents and 200 documents, respectively. In case of Saxon B and Oracle, the processing time increases linearly as the number of data increases. However, the processing time of our system is independent of the data size for searching. The result of the evaluation shows that our system outperforms so that our approach is believed to be on of the efficient approaches for managing metadata in the Set-Top Box.

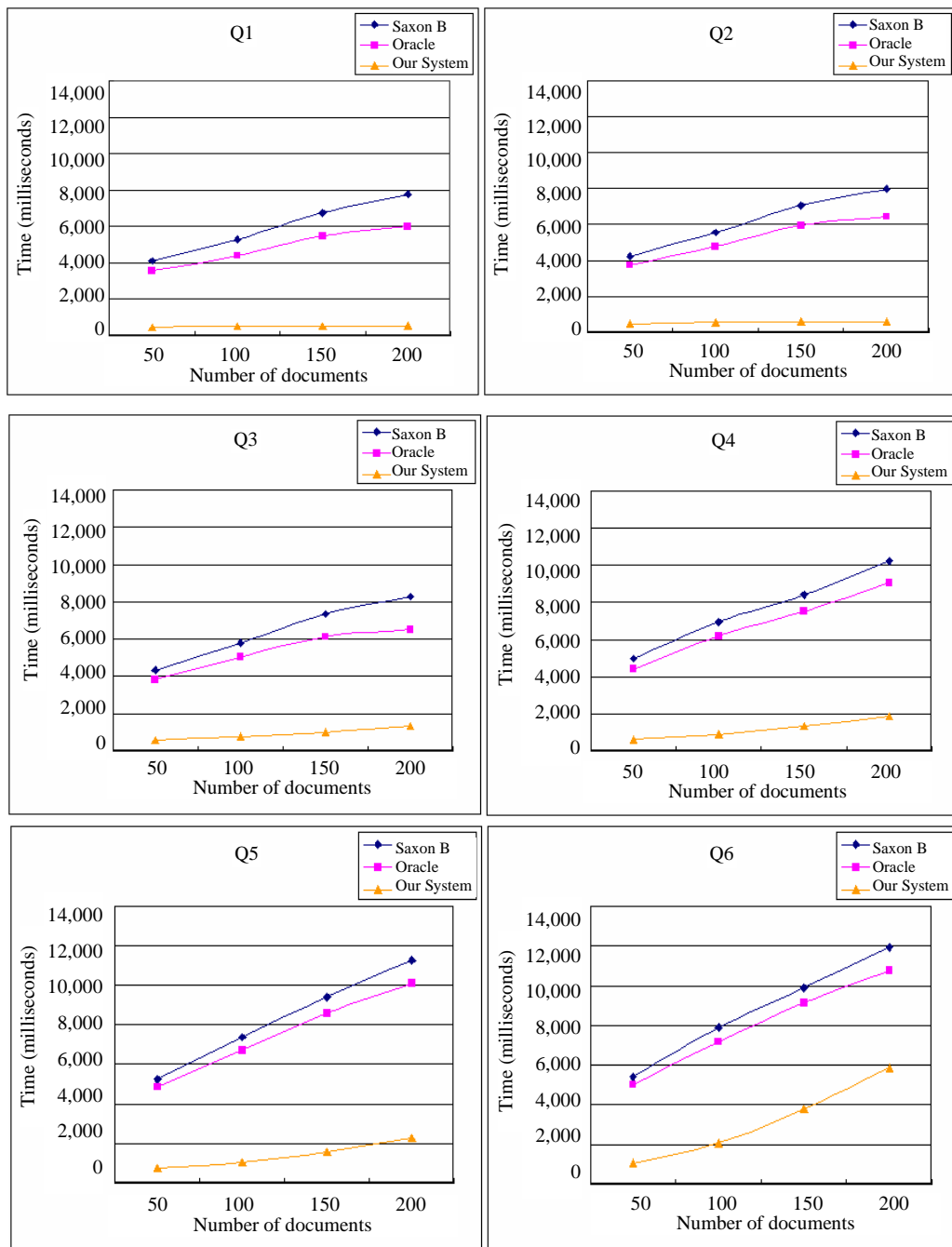


Figure 9. Performance evaluation for scalability property

6. Conclusions

In this paper, we have proposed a method for storing and searching TV-Anytime metadata for digital broadcasting based on a Set-Top Box which is low-cost and low-setting. Also we have implemented a prototype system for applying our method and evaluated our approach which seems important since our prototype system outperforms the other compared systems. Our system was developed on digital broadcast environments [18]. However our result can be applied to any XML management systems that fo-

cus on the performance of store and retrieval on low-cost environments.

7. Acknowledgement

This research is supported by MKE & IITA(08-Infrastructure-13, Ubiquitous Technology Research Center), and also by Foundation of ubiquitous computing and networking project (UCN) Project, the Ministry of Knowledge Economy (MKE) 21st Century Frontier R&D Program in Korea and a result of subproject UCN 08B3-O1-30S.

REFERENCES

- [1] S. Pfeiffer and U. Srinivasan, "TV anytime as an application scenario for MPEG-7," In Proceedings ACM Multimedia 2000, Los Angeles, October 2000.
- [2] "TV-anytime phase 1," Part 3 Metadata, ETSI TS 102 822-3-1, Vol. 1.1.1, October 2003.
- [3] TV-Anytime Forum Website: <http://www.tv-anytime.org>.
- [4] J. H. Park, J. H. Kang, B. K. Kim, Y. H. Lee, M. W. Lee and M. O. Jung, "An XQuery-based TV-anytime metadata management," Proceedings of DASFAA'05 Conference, April 2005.
- [5] H. S. Shin, "A storage and retrieval method of XML-based metadata in PVR environment," IEEE Transactions on Consumer Electronics, Vol. 49, No. 4, pp. 1136-1140, November 2003.
- [6] D. Florescu and D. Kossmann, "Storing and querying xml data using an RDBMS," IEEE Data Engineering Bulletin, Vol. 22, No. 3, 1999.
- [7] I. Tatarinov, S. D. Viglas, K. Beyer, J. Shanmugasundaram, E. Shekita, and C. Zhang, "Storing and Querying ordered XML using a relational database system", Proceedings of ACM SIGMOD Conference, June 2002.
- [8] ORACLE, "Berkeley DB introduction," <http://www.oracle.com/database/berkeley-db/>.
- [9] T. Fiebig, S. Helmer, C. C. Kanne, J. Mildemberger, G. Moerkotte, R. Schiele, and T. Westmann, "Anatomy of a Native XML Base Management System," Technical Report 01, University of Mannheim, 2002.
- [10] ORACLE, "Oracle XML data synthesis or XDS," <http://www.oracle.com/technology/tech/xml/xds/>.
- [11] SAXONICA, "SAXON XQuery Engine," <http://www.saxonica.com/>.
- [12] D. Florescu and D. Kossmann, "Storing and querying XML data using an RDBMS," IEEE Data Engineering Bulletin, Vol. 22, No. 3, pp. 27-34, September 1999.
- [13] I. Tatarinov, S. D. Viglas, K. Beyer, J. Shanmugasundaram, E. Shekita, and C. Zhang, "Storing and querying ordered xml using a RDB system," Proceedings ACM SIGMOD Conference, June 2002.
- [14] M. Yoshikawa, T. Amagasa, T. Shimura, and S. Uemura: "XRel: a path-based approach to storage and retrieval of XML documents using RDBs," Proceedings ACM Transactions on Internet Technology, Vol. 5, August 2001.
- [15] T. Grust, "Accelerating XPath location steps," Proceedings of the ACM SIGMOD Conference, pp.109-120, June 2006.
- [16] M. Barg and R. K. Wong, "A fast and versatile path index for querying semi-structured data," Proceedings of the DASFAA'03 Conference, pp. 249-256, March 2003.
- [17] S. Hidaka, H. Kato and M. Yoshikawa, "A relative cost model for XQuery," Proceedings of the SAC'07 Conference, March 2007.
- [18] K. Kang, J. G. Kim, H. K. Lee, H. S. Chang, S. J. Yang, Y. T. Kim, H. K. Lee, and J. W. Kim, "Metadata broadcasting for personalized service: A practical solution," ETRI Journal, Vol. 26, No. 5, pp. 452-466, October 2004.

Development of an Improved GUI Automation Test System Based on Event-Flow Graph

Yongzhong Lu¹, Danping Yan², Songlin Nie³, Chun Wang¹

¹School of Software Engineering, Huazhong University of Science & Technology, Wuhan 430074, P. R. China, ²School of Public Administration, Huazhong University of Science & Technology, Wuhan 430074, P. R. China, ³School of Mechanical Science and Engineering, Huazhong University of Science & Technology, Wuhan 430074, P. R. China
Email: hotmailuser@163.com

Received November 24th, 2008; revised November 30th, 2008; accepted December 1st, 2008.

ABSTRACT

A more automated graphic user interface (GUI) test model, which is based on the event-flow graph, is proposed. In the model, a user interface automation API tool is first used to carry out reverse engineering for a GUI test sample so as to obtain the event-flow graph. Then two approaches are adopted to create GUI test sample cases. That is to say, an improved ant colony optimization (ACO) algorithm is employed to establish a sequence of testing cases in the course of the daily smoke test. The sequence goes through all object event points in the event-flow graph. On the other hand, the spanning tree obtained by deep breadth-first search (BFS) approach is utilized to obtain the testing cases from goal point to outset point in the course of the deep regression test. Finally, these cases are applied to test the new GUI. Moreover, according to the above-mentioned model, a corresponding prototype system based on Microsoft UI automation framework is developed, thus giving a more effective way to improve the GUI automation test in Windows OS.

Keywords: Automated Software Testing, Graphic User Interface, Event-Flow Graph, Regression Testing, Ant Colony Optimization, UI Automation

1. Introduction

Testing GUI is a hard and monotonous labor. So far, a large number of scholars and experts have been addressing themselves to the study of related fields. In the 1970s, some scholars suggested that testing software design be modeled by finite-state machines and testing software errors be found [1]. Thereafter another researchers applied the approach to the domain of testing GUI. It was called an improved model of finite-state machines, i.e. complete interaction sequence (CIS) [2]. After having come to recognize the fact that it increasingly did not satisfy the modeling requirements of GUI automation test, experts proposed an event-flow model based on event-flow graph. They investigated a variety of automatic generation approaches to GUI test cases, which were closely connected with the adopted GUI model like above-mentioned CIS. Besides, they simultaneously presented an algorithm to check the complete testing cases [4]. And an AI planning-based approach to GUI test was employed [5,6], which utilized the partial ordering planning in the field of AI Planning and attained test cases by the goal-driven method of searching state point. During the process of generating test cases by an AI planning-based approach, hierarchical GUI test case generation is derived [7]. In addition, other contribution like Memon and his colleagues at the University of Maryland are worth attention and they have made great progress in the theories of coverage criteria

for GUI testing [8] and test oracles for GUI-based software applications [9–11]. In recent years, McMaster together with Memon presented call stack coverage for GUI test-suite reduction [12]. Moreover, AI and data mining have been applied to the relevant study of the deep regression test. Ye et al. investigated an approach to select a better way of the deep regression test by training neural network [13]. White suggested a method to use the mathematical model of Latin square to reduce case quantities [14]. Memon et al. put forward a proposal that the adaptability to software variation was improved through choosing event relationships in the deep regression test [15].

However, these approaches have not yet fully been put into practice in GUI automation test systems of industry fields for the time being, which are roughly classified into three categories: capture and replay mode, scripts-driven mode, and data-driven mode. There exists several distinct defects among them such as heavily depending on manual work, being characteristic of low adaptability to software variation, and lacking systematic management for testing cases and their coverage. Accordingly, in an effort to enhance the automation test, a more highly automated GUI testing model, which is based on the event-flow graphs, is proposed. In the model, an automation tool is first used to carry out reverse engineering for testing GUI sample so as to obtain the

event-flow graph. Then two approaches are adopted to create testing GUI sample cases. That is to say, an improved ACO algorithm is employed to establish a sequence of testing cases in the course of the daily smoke test. The sequence goes through all object event points in the event-flow graph. On the other hand, the spanning tree obtained by deep BFS approach is utilized to obtain the testing cases from goal point to outset point in the course of the deep regression test. Finally these cases are applied to test the new GUI. Moreover, according to the above-mentioned model, a corresponding prototype system based on Microsoft UI automation framework is developed, thus giving a more effective way of improving the GUI automation test in Windows OS.

Section 2 gives a brief description of GUI automation test model based on event-flow graph and also describes two types of algorithms of generating automation test cases. Section 3 depicts the development of a corresponding prototype system based on Microsoft UI automation framework. Finally, the conclusions and future work are given in Section 4.

2. A GUI Automation Test Model Based on Event-Flow Graph

In References [8, 9, 10, 11], Memon et al. presented an event-flow graph model when deeply studying the coverage criteria for GUI testing, whose purpose was to describe the mutual relationship among the object events more clearly. Thus a model, which was equipped with the most complete functions for GUI test, came into existence. But our event-flow graph model is obtained by simplifying the above model. It is actually a two-dimension vector $\langle V, E \rangle$, where V denotes event sets in GUI and E represents order relationships of event execution in GUI. Their definitions are the same as the origin. In this model, non hierarchy modeling means neglecting the process of constructing components for GUI objects, thus enhancing the automation level for GUI test. What we have to do is to find out the GUI events which are executed immediately after previous events occur in terms of GUI states. In the course of reverse engineering, every GUI event has been gone through to discover the GUI events. Based on these GUI execution events, the vector event-flow graph is established. Then aiming at the requirements of GUI automation test, an improved ant colony optimization algorithm is employed to establish a sequence of testing cases in the course of the daily smoke test. In addition, the spanning tree obtained by deep BFS approach is utilized to obtain the testing cases from goal point to outset point in the course of the deep regression test. These cases are applied to test the new GUI. These algorithms are elaborated as follows.

The improved ACO algorithm for the daily smoke test suggested in the paper defines elicitation variables and a $tabu_k$ list and takes into consideration the consanguineous combination of a max-min ant system (MMAS), an ant

colony algorithm based on an adaptive pheromone, and a type of rewards and penalty mechanism of pheromone volatilization. Its concrete formulae are concisely expressed below as subsection functions (1)–(3) and equations (4)–(6).

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{j \notin tabu_k} [\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta} & \text{if } j \notin tabu_k \\ \text{others} & \end{cases} \quad (1)$$

$$j = \begin{cases} \underset{j \in tabu}{\operatorname{argmax}} \{ [\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta \} & \text{if } q < q_0 \\ J & \text{others} \end{cases} \quad (2)$$

$$\tau_{ij}(t+1) = \begin{cases} \rho \cdot \tau_{ij}(t) + \Delta \tau_{ij}(t) & \text{vector borders on optimal paths} \\ \rho \cdot \tau_{ij}(t) - \Delta \tau'_{ij}(t) & \text{vector borders on worst paths} \\ \rho \cdot \tau_{ij}(t) & \text{others} \end{cases} \quad (3)$$

$$\eta_{ij} = (\lambda_i + 1) / (\omega_j + 1) \quad (4)$$

$$\Delta \tau_{ij}(t) = Q / \mu \quad (5)$$

$$\Delta \tau'_{ij}(t) = \mu / Q' \quad (6)$$

where the number of crunodes is the rank, the number of ants is M . $\tau_{ij}(t)$ is pheromone density of vector border (i, j). η_{ij} is a elicitation variable which denotes the elicitation factor during the solution process. λ_i is the total number of crunodes which are not accessed from the crunode i while ω_j is the total number of crunodes which are accessed from the crunode j . α, β are corresponding to a pheromone elicitation factor and a self-elicitation factor. $tabu_k$ is an accessed crunode list when next crunode is searched. q is a stochastic variable of average distribution among $[0,1]$ while q_0 is a given constant beforehand. ρ is the coefficient of pheromone volatilization. $\Delta \tau_{ij}(t), \Delta \tau'_{ij}(t)$ are pheromone increments. Q and Q' are both constants. μ is the number of repetitive crunodes, J is the result of subsection functions (1). At the beginning, initial pheromone density $\tau_{ij}(t)$ in the MMAS is equally set to maximum. When ant k moves from the crunode i at t , $P_{ij}^k(t)$ is the probability of choosing the crunode j .

According to MMAS, each pheromone density of vector border is situated in between τ_{\max} and τ_{\min} which are set in advance. If the value is bigger than τ_{\max} , it is set to be equal to τ_{\max} ; Vice versa. Such disposal is beneficial to sufficient search and getting the optimal solution. Furthermore, if the goal crunode is not accessed and its λ is equal to 1, it should be preferentially considered when another goal crunode is selected. If the algorithm is convergent, the generated event crunode sequences are the desired GUI sample test cases for testing new GUI.

The algorithm based on the spanning tree obtained by deep breadth-first search (BFS) approach for the deep regression test is described as follows.

```

ALGORITHM: BFS( $G, s$ )
  FOR ALL  $u \in V[G] - \{s\}$  { /*the initial crunode*/
    color[u] = White;
  }
  color[s] = Gray;          /*deal with the initial
                             crunode*/
   $\pi[s] = \emptyset$ ;
   $Q = \emptyset$ ;
  Enqueue( $Q, s$ );
  WHILE  $Q \neq \emptyset$  {
     $u \leftarrow \text{Dequeue}(Q)$ ;
    FOR ALL  $v \in \text{Adj}[u]$  {
      IF color[v] = White {          /*( $u, v$ ) is the
                                     tree border*/
        color[v] = Gray;
         $\pi[v].\text{Add}(u, v)$ ;
        Enqueue( $Q, v$ );
      }
      color[u] = Black;
    }
  }

```

According to the theory of the spanning tree which shows simple path is corresponding to the shortest distance [16], GUI sample event cases can be gained as follows.

```

ALGORITHM: GetTestCaseOfEvent( $\text{Vertex } v \in V$ )
  TestCase =  $\emptyset$ ;
  FOR ALL InEdge  $\in v.\text{InEdges}$  {
     $u = \text{BFSTree.Find}(\text{InEdge.SourceVertex})$ 
    TestCase[u].Add(v);
    TestCase[u].Add(u);
    WHILE u.Parent != StartVertex {
      TestCase[u].Add(u.Parent);
       $u = u.\text{Parent}$ ;
    }
    TestCase[u].Add(StartVertex);
  }

```

3. Developing the GUI Automation Test System

In the above-mentioned model, GUI hierarchy modeling is not taken into consideration and the process of components construction is neglected. Because GUI hierarchy modeling relies on the GUI logic relationships and needs manual operation, it inevitably influences the process of GUI automation test. Furthermore, with regard to GUI test case generation, an adaptive max-min ACO above based GUI test case generation algorithm is used for GUI daily smoke test, and a deep BFS based GUI test case generation algorithm is exploited for GUI deep regression test. The developing flow of GUI automation test system is shown below in Figure 1.

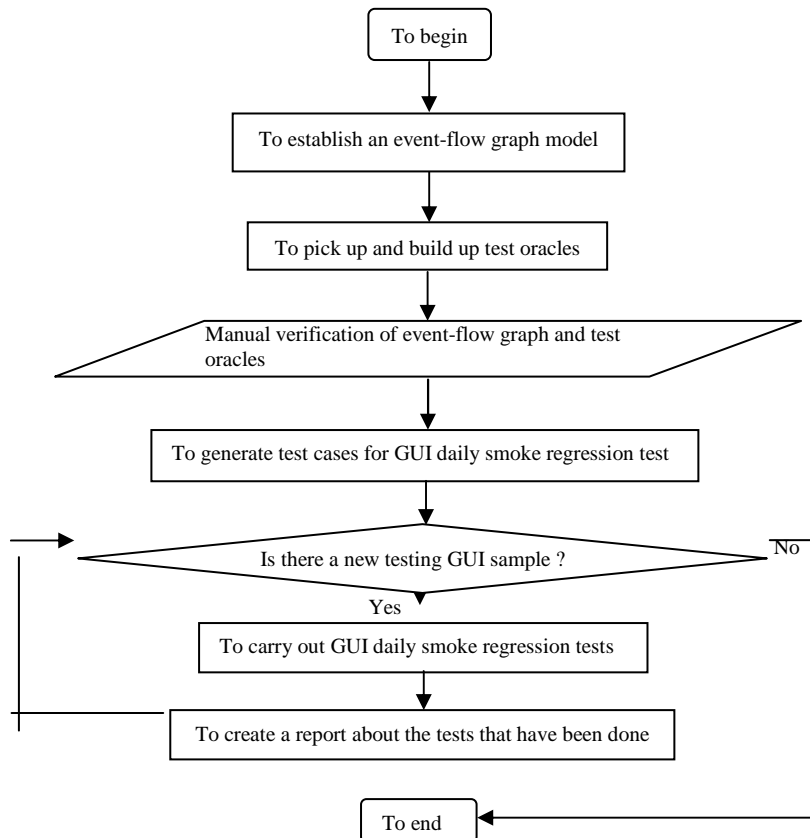


Figure 1. Developing flow of GUI automation test system

The GUI automation test prototype system is developed by taking advantage of Microsoft UI Automation frame, Visual Studio 2005, and advanced language C#. The Microsoft UI Automation frame can provide the developers with more uniform and convenient access to GUI in Windows OS than before. In the past, GUI automation operation usually requires indirect or direct usage of Windows API. Microsoft UI Automation acts as a part of Windows Presentation Foundation (WPF) in Windows SDK v6.0. It completely supports Windows Vista, Microsoft Windows XP and Windows Server 2003. It is deemed as a uniform access frame for the development of the systems based on WPF, standard Win32, Windows Form and Web UI.

The prototype system is divided into three main functional modules as follows. 1) one includes event-flow graph modeling based on reverse engineering and test oracles pick-up, 2) another one is for test case generation and report, 3) the last one is to finish testing execution and report. The output of three parts is documentary format so as to facilitate the interaction with each other and partially manual verification. Their interaction is presented in Figure 2. The hollow arrow points to the data flow direction. As Figure 2 shows, the sub-module of test oracles pick-up and another sub-module of event-flow graph modeling are used to acquire the relevant information from GUI sample, and then output test oracles and event-flow graph. Thereafter, partially

manual verification module is also exploited to inquire about whether there are some faults about GUI objects or not. After the performance, test case generation module is transferred to generate test cases for GUI daily smoke regression test. Then these cases are used for testing new GUI. Finally, testing results are passed into test report module to work out an ultimate testing document.

The first module is the most difficult one in the system because Microsoft UI Automation frame is needed to perform a dynamic automatic analysis to GUI sample. The analysis is dynamic, that is to say, the GUI information is constantly changing and there exists a extremely complex relationship between the analytic tool and GUI. This module is based on reverse engineering of GUI event-flow graph. As a result, the documentary files about vector information in event-flow graph are obtained. Figure 3 shows the interface of test oracles pick-up sub-module and event-flow graph modeling sub-module.

In test case generation module, the documentary files above are called, and then are parsed to attain hash codes of crunodes and their vector borders, and establish a vector graph objects. The above mentioned GUI test case generation algorithms are utilized to generate test cases. In particular, the function of event-flow graph plotting is designed in this module. In the process, the generally professional plotting software Graphviz is used. Figure 4 shows the interface of GUI test case generation.

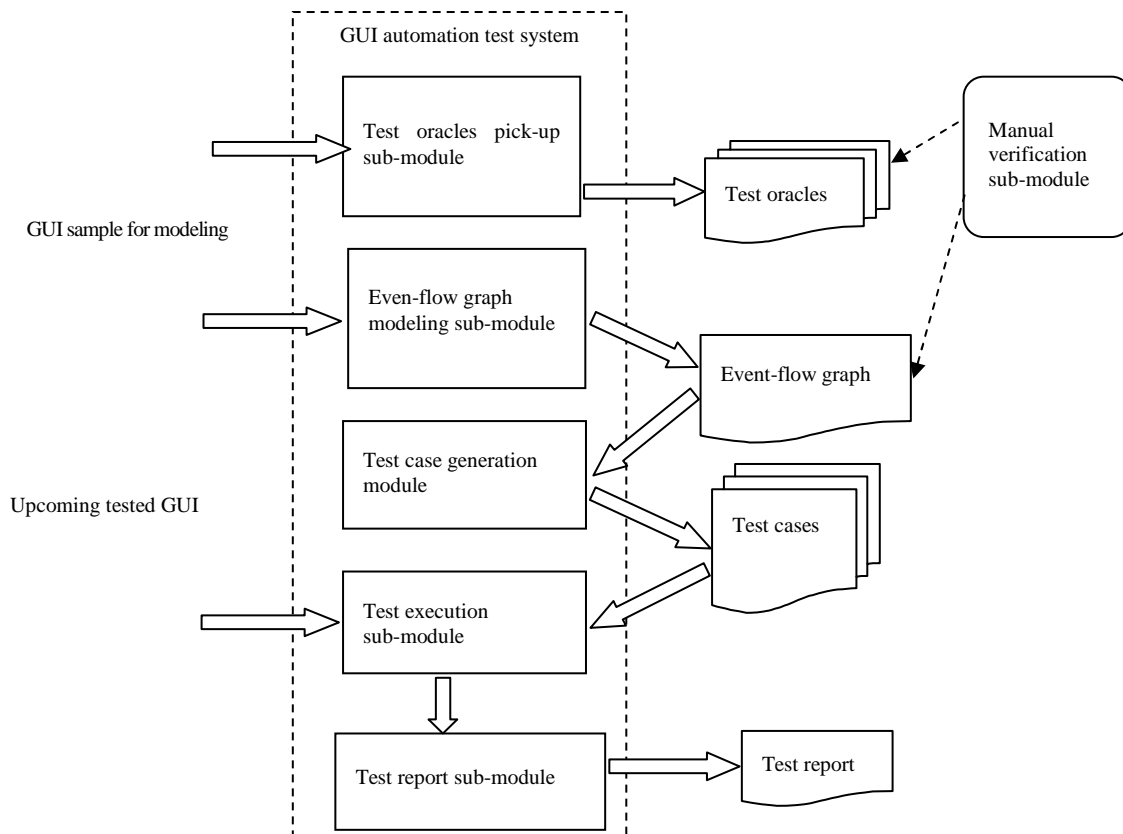


Figure 2. Interaction among three modules of GUI automation test system

In the last module of test execution and report, the required test event information can be obtained by the hash codes of new GUI. Microsoft UI Automation is used to acquire the controllers and their control modes of new GUI. The test types are selected and GUI daily smoke regression test are done. If the test is a daily smoke one, the test result is evaluated after each event is finished. If the test is a deep regression one, the test result is evaluated after the goal event is finished. Figure 5 shows the interface of GUI test execution and report.

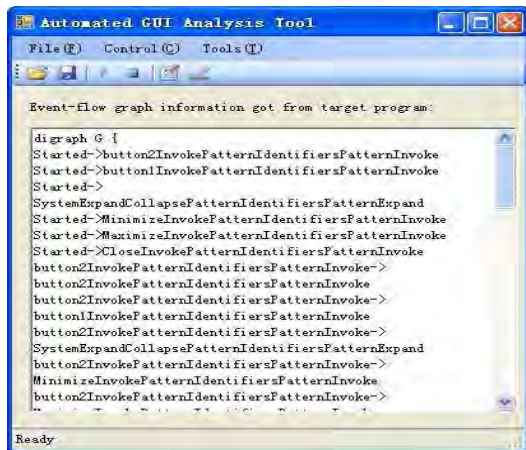


Figure 3. The interface of dealing with test oracles pick-up and event-flow graph modeling

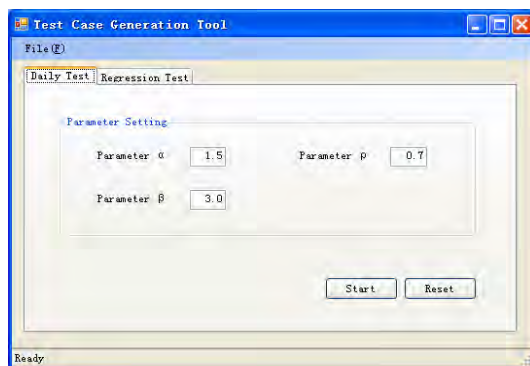


Figure 4. The interface of GUI test case generation

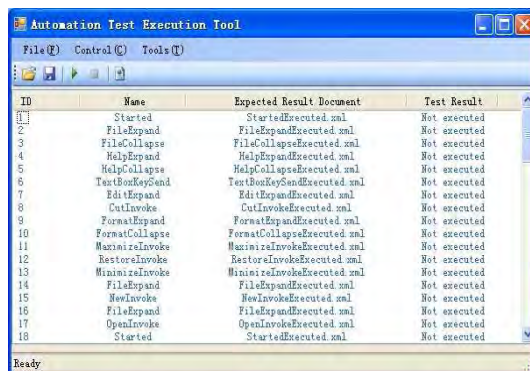


Figure 5. The interface of GUI test execution and report

4. Conclusions

Based on the event-flow graph modeling, a new GUI automation test model is presented. In the model, an improved ACO is put forward to generate test cases in the daily smoke test and a spanning tree is utilized to create test cases in the deep regression test. These test cases are generally applied in new GUI test. Moreover, a prototype system is developed on the basis of Microsoft UI Automation frame, thus giving a more effective way of improving the GUI automation test in Windows OS.

In the future, the systematic function test and contrast test with traditional GUI automation test software should be done in order to verify the validation of the model. And the adaptability of the studied system to the various GUI in other OS should be facilitated. In addition, the event-flow graph needs improving so as to solve the complex logic problem and reduce the involvement of manual verification.

5. Acknowledgement

The support from the Natural Science Foundation at Huazhong University of Science and Technology, the Natural Science Foundation in Hubei Province, and the National Natural Science Foundation in P. R. China, grant numbers 2007Q006B, 2006ABA085, 50775081, and 50675074 respectively, is gratefully acknowledged for this work by the authors.

REFERENCES

- [1] T. Chow, "Testing Software Design Modeled by Finite-State Machines," IEEE Transactions on Software Engineering, Vol. 4, No. 3, pp. 178-187, May 1978.
- [2] L. White and H. Almezen, "Generating test cases for GUI responsibilities using complete interaction sequences," in Proceedings of the International Symposium on Software Reliability Engineering, San Jose, California, USA, pp. 110-121, October 2000.
- [3] A. M. Memon, "An event-flow model of GUI-based applications for testing," Software Testing, Verification and Reliability, Vol. 17, No. 3, pp. 137-157, September 2007.
- [4] L. White, H. Almezen, and N. Alzeidi, "User-based testing of GUI sequences and their interaction," in Proceedings of the International Symposium on Software Reliability Engineering, Annapolis, Maryland, USA, pp. 54-63, November 2001.
- [5] A. M. Memon, M. E. Pollack, and M. L. Soffa, "A planning-based approach to GUI testing," in Proceedings of The 13th International Software/Internet Quality Week, San Francisco, California, USA, May 2000.
- [6] A. M. Memon, M. E. Pollack, and M. L. Soffa, "Plan Generation for GUI Testing," in Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling, Menlo Park, California, USA, pp. 226-235, April 2000.
- [7] A. M. Memon, M. E. Pollack, and M. L. Soffa, "Hierarchical GUI test case generation using automated

- planning,” *IEEE Transactions on Software*, Vol. 27, No. 2, pp. 144–155, May 2001.
- [8] A. M. Memon, M. L. Soffa, and M. E. Pollack, “Coverage criteria for GUI testing,” in *Proceedings of the 8th European software engineering conference held jointly with 9th ACM SIGSOFT international symposium on Foundations of software engineering*, New York, USA, pp. 256–267, September 2001.
 - [9] Q. Xie and A. M. Memon, “Designing and comparing automated test oracles for GUI-based software applications,” *ACM Transactions on Software Engineering and Methodology*, Vol. 16, No. 1, pp. 4–es, February 2007.
 - [10] A. M. Memon, M. E. Pollack, and M. L. Soffa, “Automated test oracles for GUIs,” in *Proceedings of the 8th ACM SIGSOFT international symposium on Foundations of software engineering: twenty-first century applications*, San Diego, California, USA, pp. 30–39, November 2000.
 - [11] A. M. Memon, I. Banerjee, and A. Nagarajan, “What test Oracle should I use for effective GUI testing,” in *Proceedings of the IEEE International Conference on Automated Software Engineering*, Montreal, Quebec, Canada, pp. 164–173, October 2003.
 - [12] S. McMaster and A. M. Memon, “Call stack coverage for GUI test-suite reduction,” in *Proceedings of the 17th IEEE International Symposium on Software Reliability Engineering*, Raleigh, North Carolina, USA, pp. 33–44, November 2006.
 - [13] M. Ye, B.Q. Feng, and Y. Lin, “Neural networks based test cases selection strategy for GUI testing,” in *Proceedings of the 6th World Congress on Intelligent Control and Automation*, Dalian, China, pp. 5773–5776, June 2006.
 - [14] L. White, “Regression testing of GUI event interactions,” in *Proceedings of the International Conference on Software Maintenance*, Monterey, California, USA, pp. 350–358, November 1996.
 - [15] A. M. Memon and M. L. Soffa, “Regression testing of GUIs,” in *Proceedings of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering*, New York, USA, pp. 118–127, September 2003.
 - [16] A. M. Memon, I. Banerjee, and A. Nagarajan, “GUI ripping: reverse Engineering of graphical user interfaces for testing,” in *Proceedings of the 10th Working Conference on Reverse Engineering*, Victoria, B.C., Canada, pp. 260–269, November 2003.

An Evaluation Approach of Subjective Trust Based on Cloud Model

Shouxin Wang¹, Li Zhang¹, Na Ma², Shuai Wang¹

¹Software Engineering Institute Beihang University Beijing, China, ²Logistics R&D Center North China Institute of Computing Technology Beijing, China

Email: shouxin_wang@126.com; lily@buaa.edu.cn; wangshuai_911@sina.com; mana82@126.com

Received November 17th, 2008; revised November 24th, 2008; accepted November 27th, 2008.

ABSTRACT

As online trade and interactions on the internet are on the rise, a key issue is how to use simple and effective evaluation methods to accomplish trust decision-making for customers. It is well known that subjective trust holds uncertainty like randomness and fuzziness. However, existing approaches which are commonly based on probability or fuzzy set theory can not attach enough importance to uncertainty. To remedy this problem, a new quantifiable subjective trust evaluation approach is proposed based on the cloud model. Subjective trust is modeled with cloud model in the evaluation approach, and expected value and hyper-entropy of the subjective cloud is used to evaluate the reputation of trust objects. Our experimental data shows that the method can effectively support subjective trust decisions and provide a helpful exploitation for subjective trust evaluation.

Keywords: Subjective Trust, Cloud Model, Trust Decision-Making

1. Introduction

With the expansion of the Internet, applications based on the internet, such as electronic commerce, online trading and networked communities are going from a closed mode to open and open mode. People and services or services providers are interacting with each other independently. Because the parties are autonomous and potentially subject to different administrative and legal domains, traditional security mechanisms based on registry, authorization and authentication have not been able to satisfy numerous web applications [1,2]. A party might be authenticated and authorized, but this does not ensure that it exercises its authorizations in a way that is expected [3]. Therefore it is important that customers be able to identify trustworthy services or service providers with whom to interact and untrustworthy ones with whom to avoid interaction. Just like Sitkin points that it is widely agreed that electronic commerce can only become a broad success if the general public trusts the virtual environment, and this means that the subject of trust in e-commerce is an important area for research [4]. Trust between the participants involved has equal importance for the nonprofit network community. It is important that we research subjective trust evaluation based on trust relation in order to ensure the customers' satisfaction in the public-oriented distributed network environment.

At present, there are two trust relations in the area [5,6], namely objective trust and subjective trust. Hypothesis-

based reasoning argumentation is a basic method in object trust research, such as BAN Logic [7] in security protocols. Subjective trust's principal component is an estimate of specific character or specific behavior level of trust objects, namely people. Trust from the principal part A to the object B means that A believes that B will definitely act in a predine or expected way under a specific circumstance [6]. This paper researches the trust decision-making of subjective trust relationships, and provides a quantitative evaluation method for subjective trust.

Many researchers have done studies on modeling and subjective trust reasoning. Papers [8,13] provide some trust evaluation and reasoning methods for probability models. Those methods don't consider fuzziness of trust itself, and their reasoning is based on pure probability models. As a result, they tend over formalize subjective trust quantification. Literatures [5,6] consider fuzziness of subjective trust, constructing subjective trust management models based on fuzzy set theory. Fuzzy set membership is a precise set description of the fuzziness but does not take the randomness into account. So, these methods lack flexibility [15]. Aiming at subjective uncertainty like randomness and fuzziness of subjective trust relationship, Beihang University advanced an approach to express trust based on a cloud model, which describes the fuzziness and uncertainty of trust [16].

Based on [16], we consider the impact of an object's reputation change with time to trust decision-making and

Thanks to the support by National Basic Research Program of China (973 project) (No. 2007CB310803)

exploited a subjective trust quantitative evaluation based on the subjective trust cloud, which preferably solves internet trust decision-making by means of analyzing historical reputation.

The remainder of this paper is organized as follows: Section 2 introduces the issue of internet trust decision-making. Section 3 describes the basic knowledge of cloud model involved in this paper. Section 4 specifies subjective trust evaluation based on cloud model and formalizes quantitatively the trust score. Section 5 shows a simulation experiment of the approach exploited in the paper and validates its validity and rationality. Finally we summarize the paper and discuss further research directions.

2. Trust Decision-making

The online trading and network communities need a set of entities providing services that they can trust. It is significant how users make a trust decision as presented in this paper. Here we call trust decision users trust subjects or subjects, entities evaluated trust objects or objects. Some large web application system, such as Amazon.com, eBay, AllExperts provide evaluation mechanisms for the reputation of subjects and objects. For objects, reputation is the evaluation of their capability, estimating intention, and capability of meeting subjects' services demands, also called objects' service satiability. In the context of this paper, we assume there is no difference in describing the trust relationship between objects trust or reputation and service satisfaction capability.

A commonly used trust decision solution is based on ratings by users, including collaborative filtering [15,16], associative retrieval [19,20], association rules [21], and Horting graphs [22]. Of these methods, collaborative filtering is the most successful. It supposes that if users grade some items similarly, they will also grade the others similarly. The basic idea of the algorithm is that the score of un-graded items given by one user are similar to ones given by the nearest neighbors of that user [17].

Recommendation system of web application provides a valuable reference for subjects' trust decision. However, the general public prefers estimation based on an object's historical reputation. Even though supported by a recommendation system, subjects are still challenged by making trust decision(s) among many recommended objects. Because the essence of subjective trust is based on subjective belief [7,8], it is random and uncertain. In addition, reputation of trust objects changes with time, which should also be quantitatively taken into account. Therefore, it is essential that Web Application Systems provide subjects with objects to select from in order to improve subject satisfaction by analyzing subjective evaluation data of the objects' history reputation.

The paper suggests a subjective trust evaluation based on cloud model, which uses history grade of reputation from subjects to objects for selecting proper objects. Our hypothesis of business environment in the paper is listed below:

- 1) There are many subjects and objects in web application systems.
- 2) Web Application Systems provide rating mechanism for evaluating objects at least.
- 3) Web Application Systems provide mechanisms for avoiding vicious and illusive evaluation.
- 4) For convenience, we use rating mechanism of five levels to explain and validate trust decision approach proposed.

3. Introduction to Cloud Model

In the reasoning process, randomness and fuzziness are usually tightly related and hard to separate [23]. Based on random and fuzzy mathematics, a cloud model can uniformly describe randomness, fuzziness, and their relationship. This chapter introduces basic knowledge of the cloud model.

DEFINITION 1: Cloud and cloud drops [24]: Assume that U is a quantitative numerical universe of discourse and C is a qualitative concept in U . If $x \in U$ is a random implementation of concept C , and $\mu(x) \in [0,1]$, standing for certainty degree for which x belongs to C , is a random variable with stable tendency.

$$\mu: U \rightarrow [0,1] \quad \forall x \in U \quad x \rightarrow \mu(x)$$

Then distribution of x in universe of discourse U is called cloud and each x is called a cloud drop.

According to definition 1, cloud has the important qualities as follows.

- 1) Cloud is the distribution of random variable X in the quantitative universal set of U . But X is not a simple random variable in the term of probability, for any $x \in U$, x has a certainty degree and the certainty is also a random variable not a fixed number.
- 2) Cloud is composed of cloud drops, which are not necessarily in any order. A cloud drop is the singular implementation of the qualitative concept. The character of concept is expressed through all drops, the more drops there are, the better the overall feature of the concept is represented.
- 3) The certainty degree of cloud drop can be understood as the extent to which the drop can represent the concept accurately.
- 4) Qualitative concept described in cloud model is reflected by many quantitative concept values and binary pairs from $\langle x, \mu \rangle$ of their certainty degree.

The general concept of a cloud model can be expressed by its three numerical characteristics: Expected value (Ex), Entropy (En) and Hyper-Entropy (He). In the discourse universe, Ex is the most representative for qualitative concept. En is a randomness measure of the qualitative concept, which indicates its dispersion on the cloud drops, and the measurement of "this and that" of the qualitative concept, which indicates how many elements could be accepted to the qualitative linguistic concept. He is a measure of the dispersion on the cloud drops, which can also be considered as the entropy of En

and is determined by the randomness and fuzziness of En .

DEFINITION 2: One-dimension normal form cloud [24]: Assume that U is a quantitative numerical universe of discourse and C is a qualitative concept in U . If $x \in U$ is a random implement of concept C , x satisfies: $x \sim N(Ex, En^2)$, $En' \sim N(En, He^2)$, and certainty of x for C satisfies the following rule:

$$\mu = e^{-\frac{(x-Ex)^2}{2(En')^2}} \quad (1)$$

Then x can be called normal form cloud in the discourse U . The paper [25] thoroughly analyzes and discusses the universe of normal form cloud in applying uncertainty representation. The cloud models involved in this paper are one-dimension normal form cloud and Figure 1 shows the graph of one-dimension normal form cloud whose numerical characteristics are $Ex=3$, $En=3$, and He is 0.01.

As defined earlier, the quantitative value of cloud drops is determined by the standard normal form distribution function. Their certainty degree function adopts a bell-shaped membership function used broadly in fuzzy set theory. As a result, normal form cloud model is a brand new model based on probability theory and fuzzy set theory, and concurrently holds randomness in the former and fuzziness in the latter.

4. Subjective Trust Evaluation Based on Cloud Model

It is important to understand and distinguish the difference of alternative trust objects from which trust decisions are made. Trust decisions in the internet environment are a process where trust subjects can distinguish the difference of reputation of alternative objects using decision constraints. Subjects choose some objects from an object set $Objs=\{obj_1, obj_2, \dots, obj_n\}$. It can generate a smaller alternative trust object set $Objs'=\{obj_1, obj_2, \dots, obj_m\}$ ($m < n$) and reduce the selection range. Decision constraints are the focus of the decision process and provide rules for distinguishing potential differences of objects' trust reputation. The formal description of trust decision process is given below as expression (2).

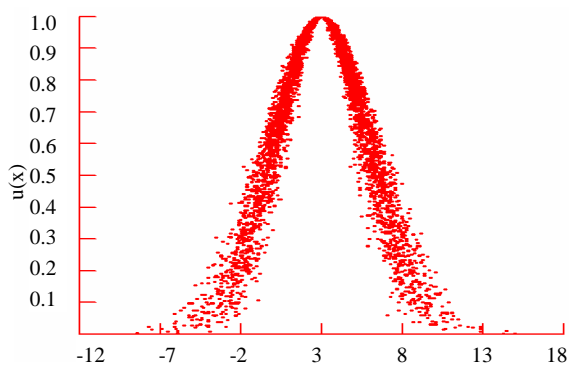


Figure 1. Cloud graph of one-dimension normal form cloud

$$Transform(Objs) \xrightarrow{\text{constraint}} Objs \quad (2)$$

A trust decision method means, subjects describe decision constraints qualitatively or quantitatively in the process of selecting objects based on analysis of potential differences in their trust reputation. In this paper, we use subjective trust cloud based on the cloud model to quantitatively describe decision constraints, and to distinguish the average level of trust reputation between multiple objects.

4.1 Subjective Trust Cloud

With a simple subjective grade mechanism and average value to calculate trust reputation, i.e., Amazon.com and OnSale and so on, evaluate a seller's trust reputation [26]. Table 1 displays five evaluation information of objects' trust reputation from Amazon.com which provides same services. Amazon provides the overall reputation of every object. The other four objects have the same overall evaluation value except A. Therefore, without other supporting information, it is difficult for subjects to make a trust decision reasonably and effectively. Statistical methods can effectively reflect randomness of subjective grade, but can't express the significance of subjective uncertainty, namely, fuzziness. As a result, it is rational to express the qualitative concept of subjective trust in a cloud model. In addition randomness and fuzziness are correlated in cloud model expression, which provides support for trust decisions more reasonably and effectively.

In this paper, we give definitions which are correlated with subjective trust cloud as follows:

DEFINITION 3: Subjective trust degree (STD) is an ordered set of number in an universal set $[0, n]$, $STD=[0, n]$. STD is composed of sequential or discrete numbers which represent a trust object's reputation and n is any positive integer. 0 and n represent the lower and upper limit of the reputation.

DEFINITION 4: Subjective trust space (STS) is an ordered set of qualitative concepts which represent the qualitative degree of trust. There can be 0 or more than one trust level standard for one STS.

DEFINITION 5: Subjective trust cloud (STC) is a subjective trust concept represented by cloud model and composed of many cloud drops. $STD=[0, n]$ is the universal set of STC, for any $e \in STS$ is a qualitative trust concept of STS, and any $x \in STD$ is a implement of e . The certainty degree of x for e , i.e., $\mu(x) \in [0, 1]$ is a random value with stabilization tendency.

Table 1. Reputation of objects from amazon.com

object	1	2	3	4	5	Aggregation
A	17	77	89	154	589	4
B	55	29	46	90	732	4.5
C	14	20	62	137	788	4.5
D	16	26	49	121	734	4.5
E	58	60	161	380	234	4.5

$$\mu: \text{STD} \rightarrow [0, n] \quad \forall x \in \text{STD} \quad x \rightarrow \mu(x)$$

Then the distribution of x on STD is defined as $\text{STC}(x)$, and every x is called subjective trust cloud drops.

The subjective trust cloud is extensible, and when the discourse space of STD is $[0, 1]$, it is equal to the trust cloud in [16]. Quantitative reputation of subjective trust cloud can be ordered value composed of any value of $[0, n]$. For STD, ordered value is composed of a set of sequential or discrete values reflecting reputation, which makes subjective trust evaluation based on cloud more pervasive. Firstly, without extra data processing, it is applicable to discrete or sequential value reputation grade mechanism. Secondly, it can effectively reflect qualitative-quantitative transformation of cloud and climbing-up of qualitative concepts. If reputation is continuous values, it reflects qualitative-quantitative transformation between subjective qualitative trust concepts and quantitative discourse. If reputation is discrete value space, it reflects climbing-up of fine granularity of concept, namely, qualitative concepts and values in discourse space form hierarchical construct of concepts.

The other characteristic of subjective trust cloud means that it doesn't necessarily require qualitative concept in trust space, namely, regulating trust grade. It evaluates overall objects' reputation by just comparing $\langle Ex, He \rangle$ which is called subjective trust character vector. It is necessary to endow its numerical characteristic with rational and significant physical meanings in the context when cloud model expresses qualitative knowledge. In this paper, we take Ex as typical value of objects' reputation, namely, average reputation level of objects. In addition, we use He to reflect decentralization degrees from objects' reputation to the average, namely, He reflects the stability of an objects' reputation. If Ex is big, then an object's ability to satisfy a subject's need is big and vice versa. If He is small, then the stability of reputation for an object is good and vice versa.

Subjective trust cloud design

The first step for a quantitative evaluation of an object's reputation is to design the STD, confirm the upper/lower limit of reputation space, and select discreteness or continuity of reputation. In this paper, we give a possible STD design, with five-grade-mechanism of Amazon.com serving as an example. When STD is a discrete space, every discrete reputation virtually can be considered as qualitative concept. STD is designed to be $[1, 2, 3, 4, 5]$ in this paper.

Generation of numerical character value of STC

Object reputation varies with time, and it associates closely with its historical reputation and time [27]. Therefore, evaluation data of subjective reputation is only effective for a given period of time. This means the further away the evaluation time from the trust decision,

the lower the effectiveness of its object reputation. In order to correctly evaluate that, we extend the cloud generation algorithm backward without certainty degree in [24], and design a weighted backward cloud generation algorithm. Based on the distance from reputation evaluation time to current trust decision time, this algorithm assigns different weights to reputation data of different times. The basic weight rule of this algorithm is, the newer the reputation data is, the bigger its weight and vice versa. We first explain the time model of reputation and basic rules for weighting.

Suppose the time model of reputation $M = \langle X, t_c, t_b, T \rangle$.

1) $X = \{x_1, x_2, \dots, x_n\}$ is the full set of historical reputation data of an object. For any x_i , $\text{Time}(x_i)$ denotes the time of reputation evaluated.

2) t_c denotes the current time of trust decision and serves as time origin. t_b denotes certain time of forward direction of time axis, and serves as time threshold for judging effectiveness of reputation.

3) $T = \{t_1, t_2, \dots, t_{m-1}\}$ is an ordered set composed of $m-1$ time values between t_c and t_b . For any t_i , $d_i = |t_i - t_c|$ is called time distance from t_i to t_c , and satisfies following constraint.

$$1) \forall d_i (1 \leq i \leq m-1) \rightarrow d_i \leq |t_c - t_b|$$

$$2) \forall d_i, d_j (1 \leq i < j \leq m-1) \rightarrow d_i < d_j$$

Based on $\text{Time}(x_i)$, t_b can separate X into two subsets, X_1' and X_2' , and they satisfy the conditions below.

$$1) X = X_1' \cup X_2', \text{ and } X_1' \cap X_2' = \emptyset$$

$$2) \forall x_i \in X_1' (1 \leq i \leq n) \rightarrow (|\text{Time}(x_i) - t_c| \leq |t_c - t_b|)$$

$$3) \forall x_i \in X_2' (1 \leq i \leq n) \rightarrow (|\text{Time}(x_i) - t_c| > |t_c - t_b|)$$

As mentioned above, t_c serves as time origin, and $|t_c - t_b|$ serves as time threshold for judging effectiveness of reputation evaluation data. The set of X is separated based on the difference of $|\text{Time}(x_i) - t_c|$ and $|t_c - t_b|$. Time distance from any element in X_1' to t_c is less than or equal to the threshold, and that of X_2' is more than the threshold. Therefore, we consider evaluation time of reputation data in X_2' , to be far away from current decision time, which can't correctly reflect the object reputation of current time. Evaluation data of object reputation is all included in X_1' .

The set T separates time interval between t_c and t_b into m sub-areas called temporal windows and marked as W_i . Temporal windows make X_1' m subsets of reputation evaluation data, $X_{t_1}, X_{t_2}, \dots, X_{t_m}$. They satisfy following conditions:

For any temporal window, $\text{Win}_{t_i} = \langle t_{\text{low}}^i, t_{\text{sup}}^i \rangle$, t_{low}^i is the lower time limit of W_{t_i} , and t_{sup}^i is the upper time limit of W_{t_i} which satisfy $|t_{\text{low}}^i - t_c| < |t_{\text{sup}}^i - t_c|$.

$\text{Win}_{t_i} = [t_{\text{sup}}^i - t_{\text{low}}^i]$ is called window length of W_{t_i}

$X_1' = X_{t_1} \cup X_{t_2} \cup \dots \cup X_{t_m}$, and

$$\forall X_{t_i}, X_{t_j} (1 \leq i \leq m, 1 \leq j \leq m) \rightarrow (X_{t_i} \cap X_{t_j}) = \emptyset$$

$$\forall y \in X_{t_i}, z \in X_{t_j} (1 \leq i < j \leq m) \rightarrow |\text{Time}(y) - t_c| < |\text{Time}(z) - t_c|$$

When we design the set of T , we should consider the time span of $[t_b, t_c]$, and quantity of reputation data in the span. T further separates X_1 into m subsets, and based on whose subject temporal windows, there is strict time sequence in $X_{t_1}, X_{t_2}, \dots, X_{t_m}$. There is equivalent weight of effectiveness for some reputation data whose time value is in the same temporal window. For any subset X_{t_i} ($1 \leq i \leq m$) of X_1 , we can assign a weight w_{t_i} , which denotes the reputation influence extent from data in X_{t_i} to that of overall results of the objects. Weights should satisfy the constraints of expressions (3) and (4). Based on these expressions, we provide a simple weight assignment method satisfying the expression (5), which is based on that, as the time distance of t_i from t_c increases, its effectiveness for a period of time fades, and we express that fading trend in the mode of descent with the same difference which is indicated by the variable *inter*.

$$\forall x_i \in X_{t_k}, x_j \in X_{t_l} (1 \leq k < l \leq m) \rightarrow (w_{t_k} < w_{t_l}) \quad (3)$$

$$\left(\sum_{i=1}^m w_{t_i} \right) = 1 \quad (4)$$

$$w_{t_{i+1}} = w_{t_i} - \text{inter} (1 \leq i \leq m-1) \quad (5)$$

After calculating the weights we can apply the weighted backward generation cloud algorithm, to calculate the subjective trust cloud values of Ex , En , He . The weighted backward generation cloud algorithm is described as follows.

Input: a set of N cloud drops, $X_1 = \{x_1, x_2, \dots, x_N\}$, and a set of cloud drops' weight, $W_i = \{w_{t_1}, w_{t_2}, \dots, w_{t_m}\}$. m indicates the number of temporal windows.

Output: (Ex , En , and He) representative of qualitative concept of N cloud drops.

Steps:

1) Calculate the weight w_i of x_i with the equation i.e.,

$w_i = \frac{w_{t_j}}{\text{numWin}_j} (1 \leq i \leq N, 1 \leq j \leq m)$. Win_j is the j th temporal window and w_{t_j} is the weight of it. $\text{num}(\text{Win}_j)$ is a function which computes the number of drops in Win_j .

2) On the basis of x_i and its weight, calculate sample mean, first-order absolute central moment, and sample variance of x_i , i.e., $\bar{X} = \sum_{i=1}^N w_i x_i$, $\sum_{i=1}^N w_i |x_i - \bar{X}|$, and

$$S^2 = \sum_{i=1}^N w_i (x_i - \bar{X})^2$$

$$3) \hat{Ex} = \bar{X}$$

$$4) \hat{En} = \sqrt{\frac{\pi}{2}} \sum_{i=1}^N w_i |x_i - \hat{Ex}|$$

$$5) \hat{He} = \sqrt{|S^2 - \hat{He}^2|}$$

4.2 Trust Decision-making

After we compute three numerical values of the subjective trust cloud, we can make trust decisions based on the foundation of its character vector. For the physics meaning of $\langle Ex, He \rangle$, we should pick objects whose Ex

is big and He is small. A formal description of the trust decision, based on the subjective trust cloud, is expressed by equation (6).

$$\text{Transform}(\text{Objs}) \xrightarrow{\langle Ex, He \rangle} \text{Objs} \quad (6)$$

But the character vectors may not accurately represent the things the trust subjects care about because they only pay attention to the result of selecting a trust objects based on some reasonable and simple rules. Therefore, similar to some existing methods [2, 28, 29, 30], it is very necessary to provide one certain approach, which can combine the Ex with He to obtain certain simple result of reputation, for trust subjects. Relying on the simple result, the most suitable object would be selected for trust subjects. Here we provide a reputation scoring method to address the issue.

As stated as above, Ex expresses the average reputation level, and He describes the decentralization degrees from reputation to the average, namely, stability of uncertainty of reputation. Hereby, for calculating quantitatively, we consider the Ex as the master value and He slave value. Reputation score is a function of Ex and He and increases with Ex and decreases with He . The formalized function of reputation score (hereafter RS) is described as $RS = Ex \times e^{-He} (7)$.

Expression 7 can represent the basic function relationship among RS, Ex and He . But in some special situations, expression 7 may have inaccurate results. To analyze these special situations, some typical cases of Ex and He are listed in Table 2.

According to expression 7, the RS is clearly better in case1 than case3. However, if there exists object A with high Ex and He , and object B with low Ex and He . Then the Ex of A may be higher than B's, but object A and B may have the same RS. In this situation, RS can not tell the fine difference of object A and B. To overcome the issue, expression 8 is introduced to amend the function of expression 7.

$$RS = Ex \times e^{-He} + \frac{b}{c} Ex (c = b + 1) \quad (8)$$

$\frac{b}{c}$ is an impact factor to adjust the computing result of RS. Expression 8 with the impact factor can distinguish the RSs among objects in case2 and case4. We can prove the validity of expression 8 as follows:

Proof: Suppose RS_a and RS_b are the reputation scores

of objects A and B. $RS_a = Ex_a \times e^{-He_a} + \frac{b}{c} Ex_a$,

$RS_b = Ex_b \times e^{-He_b} + \frac{b}{c} Ex_b$, and $Ex_a > Ex_b$.

Table 2. Table 1 four cases of EX and HE

	<i>Ex</i>	<i>He</i>
Case1	High	Low
Case2	High	High
Case3	Low	High
Case4	Low	Low

1) If $RS_a=RS_b$ then

$$Ex_a \times e^{-He_a} + \frac{b}{c} Ex_a = Ex_b \times e^{-He_b} + \frac{b}{c} Ex_b \quad \text{and} \quad \frac{Ex_b}{Ex_a} = \frac{e^{-He_a} + \frac{b}{c}}{e^{-He_b} + \frac{b}{c}} \quad (9)$$

2) Because $1 < e < 1$ and $He > 0$, so $0 < e^{-He_a} < 1$ and $0 < e^{-He_b} < 1$

3) As the result, $1 < \frac{Ex_b}{Ex_a} < \frac{b+1}{b}$

4) From the initial assumptions and the sequence of deduction steps, we can conclude that if $RS_a=RS_b$ then Ex_a approximately equals to Ex_b .

Similarly, let $\frac{Ex_b}{Ex_a} = \alpha$, then

$\alpha e^{-He_a} + \frac{b}{c} = e^{-He_b} + \frac{b}{c}$ (10). Applying natural logarithm and equation transformation to equation 8, we can get a new equation $He_a - He_b = \ln(\frac{1}{\alpha^2})$ (11). Since α is close to 1, He_a is approximately equivalent to He_b .

Computing the RS of objects by the equation 8, can limit the error into acceptable range. $\frac{b}{c}$ is used to adjust the precision of reputation score. More small the inverse of $\frac{b}{c}$, more fine difference among reputation score of objects can be distinguished.

5. Experiment and Discussion

5.1 Maintaining the Integrity of the Specifications

Because most Web Sites can't provide time of reputation and the intention of the experiment is evaluating the effectiveness of the approach in the paper, we simulated the time of reputation based on real reputation data from Amazon.com. We collected 14 objects which provide a similar service, with ratings of each service greater than 700. Table 3 shows three typical original reputation data of objects.

The simulation steps are described as follows.

1) Assume the basic time unit is a week and all reputation data has been given in past ten weeks, this means $t_b=10$ weeks.

2) Designate several different ways to divide the temporal windows

3) Calculate time weight for each temporal window based on the equation (4) and (5)

Table 3. The original reputation data of three objects

objects	1	2	3	4	5
A	264	519	496	649	967
B	571	533	504	680	363
C	424	604	903	579	756

Firstly, we divide the ten weeks into three temporal windows. The number of weeks of each window is 1, 4, and 5 respectively. Applying the weighted backward generation cloud algorithm, we can obtain the numerical characteristics of the subjective trust cloud for objects A, B, and C depicted in Table 4.

From table 4, the Ex of B is lower than that of A and C. But the Ex of A is similar to C, and their difference is only 0.07. However, the He of A is smaller than that of C. Therefore, we can say that the basic level of reputation of B is lower than others, and the stability of reputation of A is higher than B. The result shows not only that the cloud model can express the uncertainty of subjective trust, but the numerical characteristics can be used as the decision constraints for subjective trust decision-making, and indicate fine differences among objects.

Next we validate the effect of temporal window on the result of reputation evaluation based on our approach. Actually, customers or owners of web site have many optional ways to define different temporal windows. They can choose two, three, or more temporal windows, and decide the number of basic time units of each one. Table 5 gives some possible methods to divide temporal windows.

Temporal windows depicts the number of temporal windows whereas the column of Basic time unit indicates the partition of each temporal window. For example, (1, 4, 5) means the first window should contain one week, and the second and third should contain four and five weeks. The curves of Ex and He of A, C under different partitions are shown in Figure 2 below.

The red curves represent object A, and blue ones represent object C. According to the partitions of Table 5, the Ex of A is always higher than that of C, and the He of

Table 4. Reputation ranking and the numbers of STC

objects	Ex	En	He
A	3.60	1.45	0.62
B	3.13	1.51	0.62
C	3.53	1.46	0.88

Table 5. The instances of temporal windows

Serial number	Temporal windows	The number of basic time unit
1	2	(10, 0)
2	2	(1, 9)
3	2	(2, 8)
4	2	(3, 7)
5	2	(4, 6)
6	2	(5, 5)
7	2	(6, 4)
8	2	(7, 3)
9	2	(8, 2)
10	2	(9, 1)
11	10	(1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
12	3	(1, 4, 5)
13	3	(1, 2, 7)

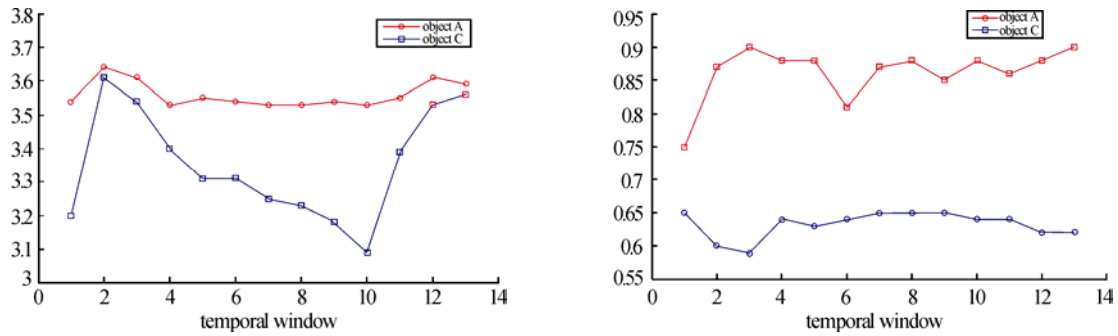


Figure 2. Ex and He curves of A and C

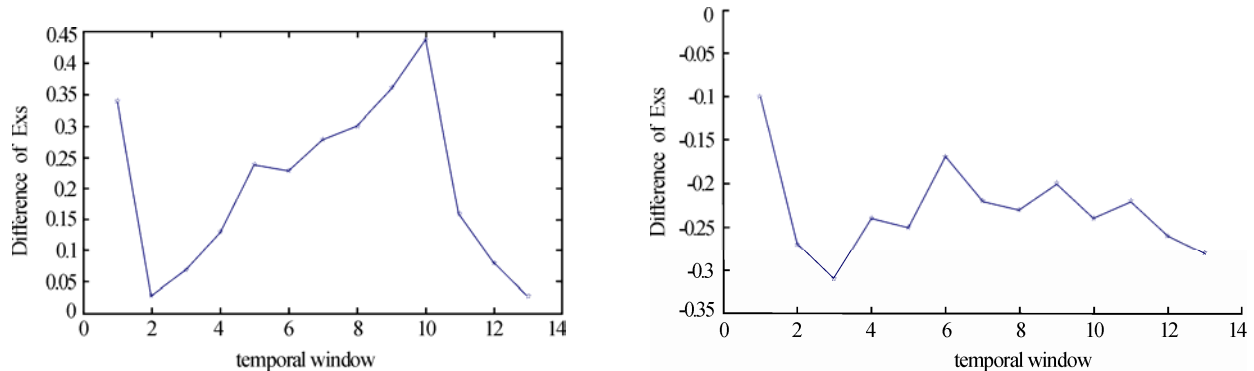


Figure 3. Curves of difference of Ex and He for A and C

A is smaller than C. Therefore, we can conclude that different partition methods don't change the result of reputation evaluation based on the subjective trust cloud. But different partitions can affect the precision of reputation evaluation. To exhibit this, the curves showing the difference of Ex and He of A, and C are depicted in Figure 3.

In Figure 3, the difference of Ex reaches the maximal value at the tenth partition, and the minimum at the second partition. However, the maximum and minimum of He are achieved at the first and third partition. So the trend of the two curves is not absolute consistent. We believe the distribution of reputation data may be what causes the difference under different partitions. Additionally, from Figure 2 and 3, the difference of Ex of A and C is more than zero, while their He difference is less than zero. Although different partitions may result in dissimilar evaluation, we can obtain the same conclusion which is consistent with that from Figure 2. That is the result of reputation evaluation does not change with the partition method.

5.2 Reputation Scoring Function

Based on the values of Ex and He in table 4, we apply the reputation scoring function mentioned in section 4.2 to compute the quantitative reputation scores of trust objects. Then the RSs can be calculated and the graphs of the RSs

under different $\frac{b}{c}$ is shown in Figure 4.

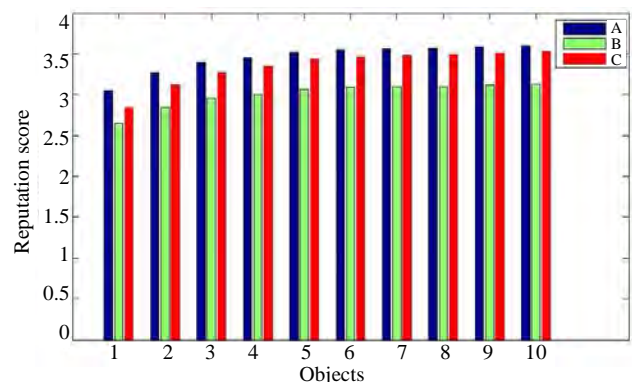


Figure 4. Reputation scores of trust objects A, B and C

There are ten groups of columniations in Figure 4. The value of c of each group from left to right is 3, 5, 8, 7, 21, 31, 41, 51, 101, 1001. The RS changes clearly from 3 to 21, but these ones between 31 and 1001 are very similar. So it is not necessary to give c a high value. On the other

hand, $\frac{b}{c}$ can control the precision to tell difference of

RSs. Actually RS of reputation may be in the range from $\frac{Ex}{c}$ to Ex . At the same time, we could find that different

c would not affect the order of reputation scores for objects A, B, and C. From the view of reputation scores, object A may be the final one selected by trust subjects. The choice result based on reputation score is consistent

with that one based on Figure 2 and 3, but more simple and suitable to trust subjects.

6. Conclusions

Cloud model overcomes the limit of fuzzy set theory which represent fuzzy concept with an accurate and sole membership degree. We proposed an evaluation approach of subjective trust based on subjective trust cloud. The approach combines *Ex* with *He* of subjective trust cloud to evaluate the randomness and fuzziness of subjective reputation. We validated our approach with a simulation experiment and showed the effectiveness of the approach. Our approach needs time of reputation. However, most Web Sites don't provide this data. But with development of business and cooperation on the Internet, especially with more attention put on satisfaction of general public, we believe that the evaluation of reputation change will be a novel and effective approach to assist end-users in trust decision-making. Furthermore there is still a need for significant research in this field, such as how to extend the approach to apply in the other related field, how to design and validate other weighting methods of reputation, how to combine subjective with objective trust data to make trust decisions, find the reasonable law and rules to design temporal windows and so on.

REFERENCES

- [1] M. Blaze, J. Feigenbaum, J. Ioannidis, et al., "The role of trust management in distributed systems security," *Secure Internet Programming: Issues for Mobile and Distributed Objects*, Berlin: Springer-Verlag, pp. 185–210, 1999.
- [2] R. Khare and A. Rifkin, "Trust management on World Wide Web," *World Wide Web Journal (S1085-2298)*, Vol. 2, No. 3, pp. 77–112, 1997.
- [3] B. Yu and M. P. Singh, "An evidential model of distributed reputation management," *International Conference on Autonomous Agents, Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, pp. 294–301, 2002.
- [4] S. B. Sitkin, D. M. Rousseau, R. S. Burt, and Camerer, "Trust in and between organizations," *Academy of Management Review*, Vol. 20, No. 3, pp. 387–620, 1998.
- [5] T. Wen and C. Zhong, "Research of subjective trust management model based on the fuzzy set theory," *Journal of Software*, Vol. 14, No. 8, pp. 1401–1408, 2003.
- [6] T. Wen, H. Jianbin, and C. Zhong, "Research on a fuzzy logic-based subjective trust management model," *Computer Research and Development*, Vol. 42, No. 10 pp. 1654–1659, 2005.
- [7] M. Burrows, M. Abadi, and R. M. Needham, "A logic of authentication," *The Royal Society of London, DEC Systems Research Center, Technical Report*, pp. 39, 1989.
- [8] M. Blaze, J. Ioannidis, and A. D. Keromytis, "Experience with the keynote trust management system: Applications and future directions," *iTrust*, New York: Springer, pp. 284–300, 2003.
- [9] M. Blaze, J. Feigenbaum, and A. D. Keromytis, "KeyNote: trust management for public-key infrastructures," B. Christianson, B. Crispo, S. William, et al., eds, *Cambridge 1998 Security Protocols Intl. Workshop*, 1998.
- [10] T. Beth, M. Borcherdig, and B. Klein, "Valuation of trust in open networks," *Proceedings 1 European Symposium on Research in Security (ESORICS)*, Berlin: Springer-Verlag, pp. 3–18, 1994.
- [11] R. Yahalom, B. Klein, and T. H. Beth, "Trust relationships in secure systems—A distributed authentication perspective. Proceedings 1993 IEEE Symposium on Research in Security and Privacy 1. Los Alamitos: IEEE Computer Society Press, pp. 150–164, 1993.
- [12] S. K. Liu and X. T. Liu, "A new method of elevation of confidence level of large-scale perplexing simulation system," *Journal of System Simulation*, Vol. 13, No. 5, pp. 666–669, 2001.
- [13] A. Jøsang, "A logic for uncertain probabilities," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems (S0218-4885)*, Vol. 9, No. 3, pp. 279–311, 2001.
- [14] Y. Wei, J. S. Li, and P. L. Hong, "Distributed peer-to-peer trust model and computer simulation," *Journal of System Simulation*, Vol. 18, No. 4, pp. 938–942, 2006.
- [15] X. Y. Meng, "Research and implement on trust model and trust evaluation system based on cloud model [MS. Thesis]," BeiHang University, 2007.
- [16] X. Y. Meng, Zhang Guang-Wei, et al., "Research on subjective trust management model based on cloud model," *Journal of System Simulation*, Vol. 19, No. 14, pp. 3310–3317, 2007.
- [17] G. W. Zhang, D. Y. Li, et al., "A collaborative filtering recommendation algorithm based on cloud model," *Journal of Software*, Vol. 18, No. 10, pp. 2403–2411, 2007.
- [18] G. W. Zhang, J. C. Kang, et al., "Context based collaborative filtering recommendation algorithm," *Journal of System Simulation*, Vol. 18, No. 2, pp. 595–601, 2006.
- [19] H. Zan, C. Hsinchun, and Z. Daniel, "Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering," *ACM Transactions on Information Systems*, Vol. 22, No. 1, pp. 116–142, 2004.
- [20] B. Thiesson, C. Meek, D. M. Chickering, and D. Heckerman, "Learning mixture of DAG models," *Microsoft Research, Technical Report, MSR2TR297230*, 1997.
- [21] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Analysis of recommendation algorithms for E-commerce," In: *Proceedings of the 2nd ACM Conference on Electronic Commerce*, New York: ACM Press, pp. 158–167, 2001 <http://www.research.ibm.com/iac/ec00/>.
- [22] C. C. Aggarwal, J. Wolf, K. L. Wu and P. S. Yu, "Horting hatches an egg: A new graph-theoretic approach to collaborative filtering," In: *Proceedings of the 5th ACM*

- SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, pp. 201–212, 1999.
- [23] D. Y. Li, C. Y. Liu et al., “Artificial intelligence with uncertainty,” *Journal of Software*, Vol. 15, No. 9, pp. 1583–1594, 2004.
- [24] D. Y. Li and Y. Du, “Artificial intelligence with uncertainty,” Chapman & Hall/CRC Taylor & Francis Group, 2008.
- [25] D. Y. Li and C. Y. Liu, “The universality of normal cloud model,” *Engineering Science*, Vol. 6, No. 8, pp. 28–34, 2004.
- [26] G. Zacharia and P. Maes, “Trust management through reputation mechanisms,” *Applied Artificial Intelligence*, Vol. 14, pp. 881–907, 2000.
- [27] E. M. Maximillen and M. P. Singh, “Conceptual model of web services reputation,” *ACM SIGMOD*, Special section on semantic web and data management, Vol. 31, No. 4, pp. 36–41, 2002.
- [28] L. Z. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, and Q. Z. Sheng, “Web engineering: Quality driven web service composition,” *ACM Press Proceedings of the twelfth international conference on World Wide Web*, May 2003.
- [29] S. Majithia, A. S. Ali, O. F. Rana, and D. W. Walker, “Reputation-Based Semantic Service Discovery,” *Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2004, WET ICE 2004, 13th IEEE International Workshops on, pp. 297–302, 2004.
- [30] L. Z. Zeng, B. Benatallah, et al., “QoS-Aware middleware for web services composition,” *IEEE Transactions on Software Engineering*, Vol. 30, No. 5, pp. 311–327, 2004.

Motif-based Classification in Journal Citation Networks

Wenchen Wu¹, Yanni Han¹, Deyi Li²

¹(State Key Lab of Software Development Environment, Beihang University, Beijing, 100083, China), ²(Institute of Electronic System Engineering, Beijing, 100039, China)

Email: wuws@nlsde.buaa.edu.cn, ziqinli@public2.bta.net.cn, libra_hyn@sina.com

Received November 16th, 2008; revised November 20th, 2008; accepted November 27th, 2008.

ABSTRACT

Journals and their citation relations are abstracted into journal citation networks, basing on CSTPC journal database from year 2003 to 2006. The network shows some typical characteristics from complex networks. This paper presents the idea of using motifs, subgraphs with higher occurrence in real network than in random ones, to discover two different citation patterns in journal communities. And a further investigation is addressed on both motif granularity and node centrality to figure out some reasons on the differences between two kinds of communities in journal citation network.

Keywords: Motif, Classification, Journal Citation Networks

1. Introduction

As an effective method, complex networks have been widely used to describe many complicated real world systems. It can be regarded as the topology abstract of many real complex systems, whose structure do not rely on node position or edge form, but with two essential attributes-small-world [1] and scale-free [2].

Though many networks present common global characteristics, they could have entirely different local structures. Recent researches indicate that network motifs, interconnected patterns occurring in numbers that are significantly higher than those in identical randomized networks, may be the “simple building blocks” in complex networks [3]. The concept and applications of motifs are first appeared in biological field. They present in biological systems as characteristic modules to carry out some certain kind of functions. For example, the same motifs, defined as feed-forward loops, have been found in organisms from bacteria and yeast [4], to plants and animals [5,6]. This kind of motifs plays an important role of persistence detectors, or pulse generator and response accelerators. These kinds of research results always make some direct biology meanings [7]. Besides, with certain iterations, many small, highly connected topologic motifs could combine in a hierarchical manner into larger but less cohesive modules [8].

Ron Milo proposed two concepts in 2002, which are shown below to find motifs in networks. And then they gave the concept of “superfamilies” in 2004 and also the significance profile (SP) method to compare the local structure between different kinds of complex networks [9]. The result shows that networks from different fields can share similar characteristic of local structures.

- Z-Score, valuing the statistic importance of each

network motifs

$$Z_i = \frac{N_{real_i} - \langle N_{rand_i} \rangle}{std(N_{rand_i})}$$

- P-Value means the probability of network motifs appearing in a randomized network an equal or greater number of times than in the real network.

Milo and his fellows also published the motif detection software, named MFinder in the homepage of Uri Alon lab. In MFinder, the subgraphs need to satisfy the default settings to make themselves network motifs, in which their Z-Score should bigger than 2, and P-Value should less than 0.05. Figure 1 shows a motif detection in real network and random network respectively by Ron Milo.

The reminder of this paper is organized as follows. Section 2 outlines the construction and essential attributes of journal citation networks. Section 3 presents the degree analysis of the networks. Section 4 analyses the motif structures and citation patterns in journal communities, and Section 5 concludes the whole work and discusses future research directions.

2. Construction Principles and Attribute Analysis

This article obtains the original data from a project led by the Institute of Scientific and Technical Information of China in 2004 [10]. The project formed both the citing and cited matrixes of each journal, which is embodied in China Scientific and Technical Papers and Citations (CSTPC) database from year 2003 to 2006 respectively. The journal citation networks are constructed according to those matrixes.

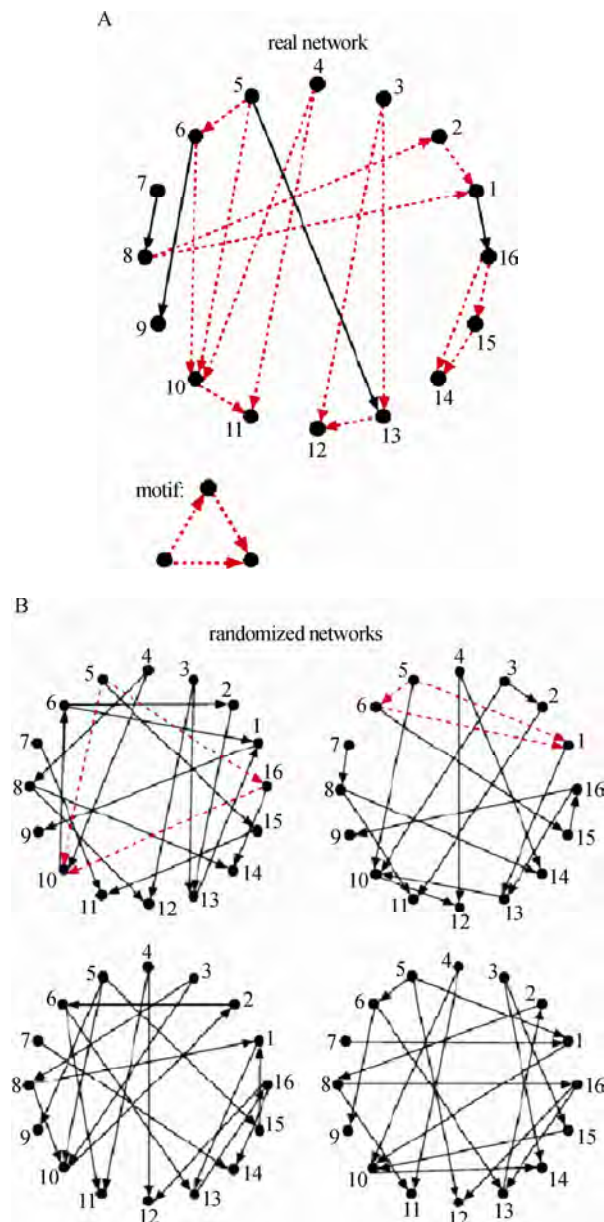


Figure 1. Motif detection in networks

In this paper, we define a journal citation network as follows: each journal expresses as a node of network, if one journal cites others, an edge is added beginning with

this journal and ending with the journal it cites. Otherwise, if one journal is cited by other ones, then add an edge beginning with the journal which cites it and ending with this journal. After sorting one-year journals this way, a directed journal citation network can finally be formed.

This article presents many essential attributes in journal citation networks of these four years. For instance, network connectivity, network diameter, average path length and also average clustering coefficient. Table 1 shows some fundamental statistic information. It can be found that network scale grows steadily from the year 2003 to 2006, except a sharply edge decrease in the year 2005. According to a further investigation, this phenomenon has something to do with a limited threshold in the original datasets, which is set up to filtrate the noise data. The average degree can explain average citation times between journals. It shows that the citation is more positive in 2006 than other years. Meanwhile, network diameters are no bigger than six in the year 2003, 2004 and 2006, and these networks also have big average clustering coefficient, which indicate typical small-world characteristic commonly in complex networks.

3. Degree Analysis

Degree is a simple but important definition to describe node attributes, which can reflect some network characteristics intuitively. When it comes to directed journal networks, a node's outdegree is the number of journals it cites, and its indegree is the number of journals citing it. Figure 2 shows the indegree and outdegree distributions of these four years. It is obvious that the nodes whose indegree or outdegree are bigger than 10, are in accordance with power-law distribution, which means a typical scale-free characteristic. Most nodes of small degrees have few cited or citing relations, but in contrary, large quantities of citation relations are held in only a few nodes. Particularly, though some nodes with really small degrees are not accordance with power-law distribution, their citations are totally rare when comparing with the entire network scale. To some extent, this kind of journals is the so-called fringe journals, and it does not play a vital part on the distribution characteristic of globe network.

Table 1. The statistical data of fundamental attributes

	2003	2004	2005	2006
node number	1577	1659	1658	1787
link number	32823	42909	25923	47470
ratio L/N	20.81357	25.86438	15.6351	26.56407
maximum indegree	211	264	255	288
maximum outdegree	98	84	124	245
average degree	36.27774	44.51718	27.22799	47.6911
network diameter	5	6	8	6
average path length(reachable)	3.47	3.242	4.073	3.782
average path length(bidirectional)	2.709	2.616	2.969	2.642
average clustering coefficient	0.238	0.246	0.269	0.302

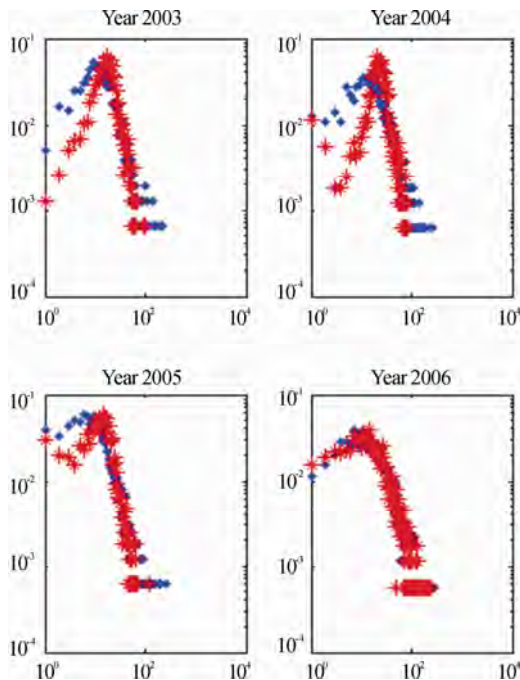


Figure 2. The indegree and outdegree distribution

In order to give a further look on the degree characteristic of these journals, Figure 3 shows the four-year correlations between indegrees and outdegrees in journal networks, in which each point corresponds to a node, and the x-position is determined by the node's indegree, the y-position corresponds to its outdegree. We can find that most nodes in journal network have significant distances with indegree and outdegree value. They are either with larger indegree but smaller outdegree, or vice versa. Only few nodes have both large indegree and outdegree. This characteristic is especially obvious in the year 2006.

It presents that nodes with large indegree have a good opportunity to be retrieved by SCI or EI, such typical examples including Chinese Science Bulletin, Chinese Journal of Computers, and etc. This kind of journals usually has a great influence in domestic journals of the same kind, which lead to a positive citations to them. But when it refers to their comparatively smaller outdegree, we believe it has a strong probability these influential journals prefer to make citations with those international journals. What have analyzed above indicates one citation characteristic of Chinese journal networks, that is journals retrieved by SCI/EI generally have a highly inclination to be cited, but with low positivity to cite other non-core journals in contrast.

4. Motif Structure in Journal Citation Networks

4.1 Motif Structure in Communities

It has been mentioned in the previous article that it is important to research on network local topology structure and generate mechanisms. In recent years, people find a clustering characteristic in complex networks [11-16].

Newman proposed the concept of community structure to indicate that an entire network is comprised of some communities or clusters. Nodes are joined together in tightly-knit groups, between which there are only looser connections. The community structure reflects high clustering and modularized characteristics. Many real networks, such as biology network, WWW network and social network have all been proved had obvious community structures. This article makes an analysis on 2004 journal citation network, and also finds the typical community structure in this network.

For the category differences, the citation times between different kinds of journals are extremely different. For example, there are only no more than ten times citations between class of physic and traffic, but thousands of times citation between all kinds of medical journals, such as pharmacy, clinical medicine and traditional Chinese medicine, etc. In principle, tight citation correlations make journals assemble in the same community, while loose citation correlations make journals separate into two communities. Through the designed experiment, journals of the same category or several similar categories generally appear in the same community with the partition of the whole journal network into twenty different communities in all.

In the following work, this article analyzes the different citation relations between those different communities. Motif kind presents in an exploding way with the increase of node number. For example, there are 13 kinds of motif with three nodes, while the number of kind rises to 199 with four-node motifs. Since journal network belongs to sociology field, and it is found that social networks are more likely to contain triangle relations. Therefore, in this paper, the research is mainly outspread on the granularity of three-node motifs.

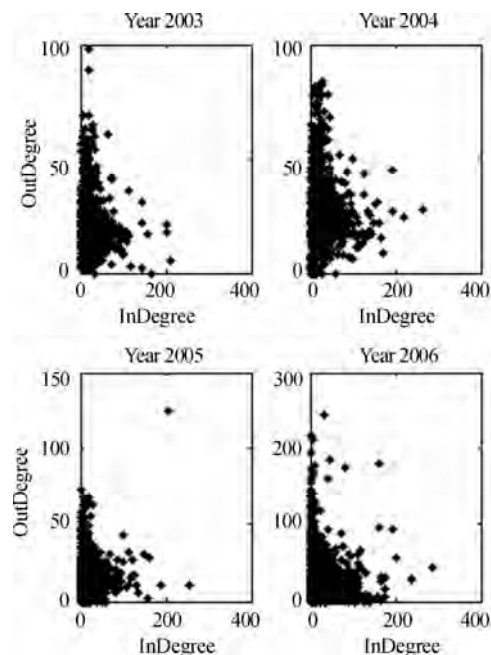


Figure 3. The indegree and outdegree correlations














According to the concrete meanings in journal citation networks, these thirteen motifs are classified into two kinds: one named “unidirectional citation clusters”, comprising with motif ID36, ID12, ID6, ID38 and ID140. This kind of motifs have a common characteristic, that is none of them contains any bidirectional edges, which means any pairs of nodes in these three-node motifs have no mutual citation relationships. To be contrast, the other kind named “mutual citation clusters”, with the motif members of ID164, ID14, ID78, ID166, ID174, ID46, ID102 and ID238, in which there could be one or even more bidirectional edges. In other words, it has at least one mutual citation relationship between the three nodes.

It indicates in Figure 3 the unidirectional citation clusters play an absolutely dominant part in journal networks, proving Chinese journal networks are more inclined to display unidirectional citation correlations. On the other hand, the occurrence of mutual citation clusters is much lower, but their Z-Score [13] values are generally much higher than motifs belonging to the unidirectional citation clusters. Z-Score is a certain variable to weigh the statistical significance in real networks with a comparison

to the corresponding randomized networks. Generally speaking, the higher Z-Score a motif has, the more significant for it to present typical characteristics in a network. Considering in the journal citation network, the mutual citation clusters’ high Z-Score value can partly illuminate the mutual citation pattern is a special pattern occurring in journal networks.

Figure 4 shows motif frequency distribution of partial communities, according to which we can classify these communities into two kinds. In the first kind, the motifs lying in the frontal part of coordinate show a higher frequency compared to the second kind, while the motifs lying in the latter part of coordinate have a lower frequency. The second kind displays in a completely opposite way. To a further analysis, the first kind communities are commonly large in node scale, for example, the Medical Sciences community has 437 nodes; the Biological and Agricultural community has 184 nodes. The second kind communities have relatively small node scale, with only 25 nodes in Light Industry & Textile community and 37 nodes in Chemical Sciences community.

Table 3. The Motif Frequencies in Several Network Communities (2004)

							
Civil & Water	10.62%	12.56%	18.66%	16.45%	14.24%	3.84%	6.10%
Mathematical Sciences	10.20%	10.41%	10.95%	11.82%	12.04%	4.77%	9.76%
Biology & Agriculture	9.68%	26.85%	16.12%	19.26%	8.48%	3.08%	5.97%
Light Industry & Textile	3.55%	9.22%	10.28%	28.37%	9.93%	4.26%	1.06%
Traffic Related	9.24%	11.14%	17.08%	19.97%	14.27%	7.01%	3.80%
Mechanical Engineering	6.38%	16.53%	10.55%	25.48%	9.38%	6.04%	3.72%
Chemical Sciences	6.20%	11.32%	7.91%	22.97%	13.02%	10.02%	2.66%
Electronic Info. & Computer	9.79%	27.90%	15.01%	17.39%	8.35%	2.32%	7.16%
Geological & Geophysical	4.87%	11.99%	8.97%	23.53%	9.54%	7.35%	3.88%
Material Sciences	5.51%	22.65%	9.32%	26.60%	5.63%	5.01%	5.63%
Comprehensive	24.65%	23.27%	30.71%	8.51%	7.10%	0.34%	3.28%
Medical Sciences	14.82%	38.36%	19.29%	11.16%	6.62%	0.92%	4.18%
Physical Sciences	6.17%	12.88%	11.99%	24.34%	10.94%	7.94%	4.06%
							UCC
Civil & Water	1.00%	3.57%	2.36%	3.89%	5.10%	1.63%	48.92%
Mathematical Sciences	0.98%	6.94%	5.75%	4.45%	9.76%	2.17%	42.30%
Biology & Agriculture	0.38%	2.96%	1.85%	1.79%	2.73%	0.82%	59.01%
Light Industry & Textile	1.16%	7.80%	0.71%	1.06%	11.70%	12.06%	25.27%
Traffic Related	0.35%	2.56%	2.48%	2.97%	6.19%	2.15%	41.60%
Mechanical Engineering	0.27%	5.57%	2.16%	2.88%	6.98%	3.98%	37.44%
Chemical Sciences	0.33%	5.04%	2.59%	2.52%	9.41%	6.07%	28.42%
Electronic Info. & Computer	0.59%	4.98%	2.18%	1.18%	2.19%	1.23%	60.44%
Geological & Geophysical	0.06%	6.41%	2.75%	3.86%	9.74%	6.53%	29.77%
Material Science	6.38%	2.32%	2.13%	6.82%	1.94%	0.08%	49.50%
Comprehensive	0.46%	0.17%	0.53%	0.19%	0.57%	0.31%	82.37%
Medical Sciences	0.88%	1.35%	1.14%	0.58%	0.91%	0.51%	77.53%
Physical Sciences	3.88%	3.17%	3.88%	7.05%	2.82%	none	38.98%

*UCC means unidirectional citation clusters

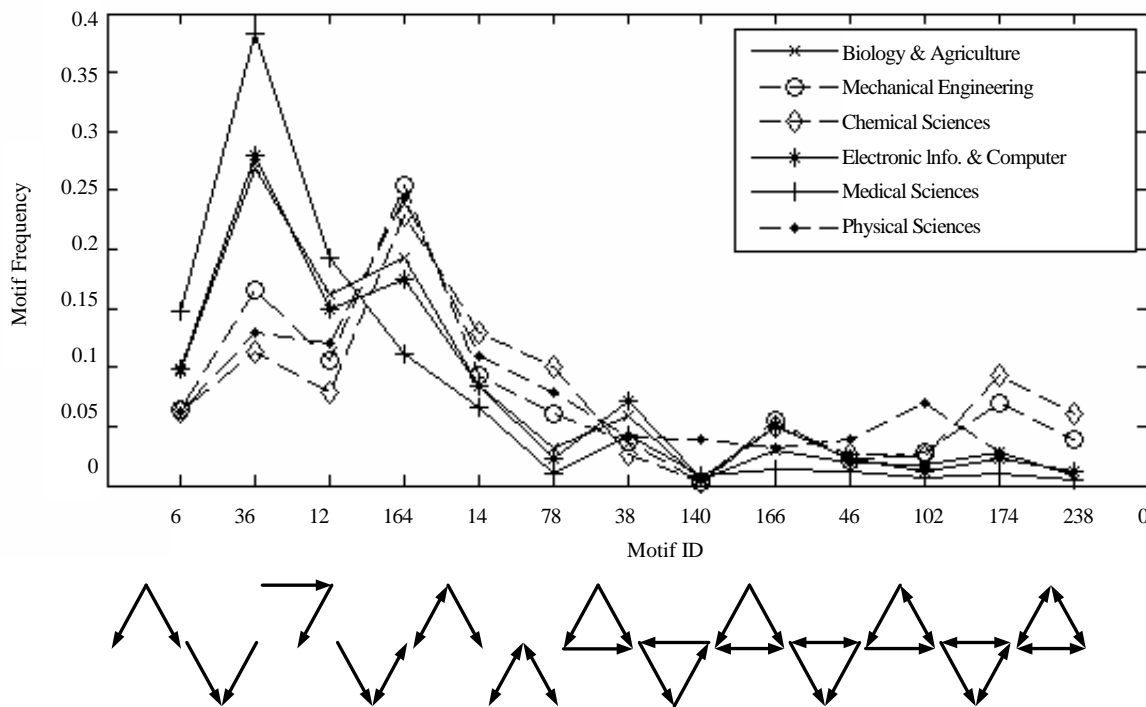


Figure 4. The comparison of motif frequencies in communities

Considering with nodes connection principles in the twenty communities respectively, it is found that in the first kind of communities, the sum frequencies of motifs in unidirectional citation clusters exceed 50%, meaning a dominance of unidirectional citation patterns. A few “hub nodes” have much larger indegree, and other nodes are inclined to connect to these nodes. However, these “hub nodes” always have very few citing connections with other nodes, even containing in the same community.

For the second kind of communities, the sum value of motif frequencies belonging to the mutual citation clusters is more than 50%, making a dominance of mutual citation pattern. We can see from above analysis that most nodes in this kind of community play a common role with no citation inclination in them.

When considering node or edge as the basic granularity, one characteristic of community structure is the loose connections between two communities. Then what characteristics it will show when take three-node motifs as the basic granularity? For a further investigation, we also take a research on motif constitution and citation patterns between these twenty communities. We find that citation pattern between communities are entirely inclined to the unidirectional pattern, meanwhile the frequency of unidirectional citation clusters is more than 70% between most communities. This statistical data is even up to 100% between biology and agriculture community and electronic information and computer community.

Meanwhile, it is also shown the frequency of the unidirectional citation clusters between any two

communities is generally much higher than the corresponding frequency in both two communities. For example, the electronic information & computer community and medical community both have an inclination to unidirectional citation pattern with the frequency of unidirectional citation cluster 60.44% and 77.53%, respectively. But this frequency rises up to 91.28% between these two communities.

4.2 Node Centrality in Communities

The structure of complex networks is typically characterized in terms of heterogeneous and topology differentiate of nodes. Take node centrality in different communities into consideration. Based on the classical centrality measures, here this paper mainly discusses degree centrality and closeness centrality. The former reflect the numbers of links incident upon a node, while closeness centrality defines as the reciprocal of geodesic distance between nodes. Since the lower closeness value a node has, the higher distance it reaches other nodes, here we take two typical networks from the two kinds of communities. One is electronic information & computer community as the unidirectional citation pattern community and the light & textile industry community as the example of mutual citation pattern. Figure 5 shows the frequency distribution of node centrality in the electronic information and computer community

It is shown that nodes with large indegrees generally corresponding to small outdegrees, and the ones with large outdegrees turn out to have small indegrees. The

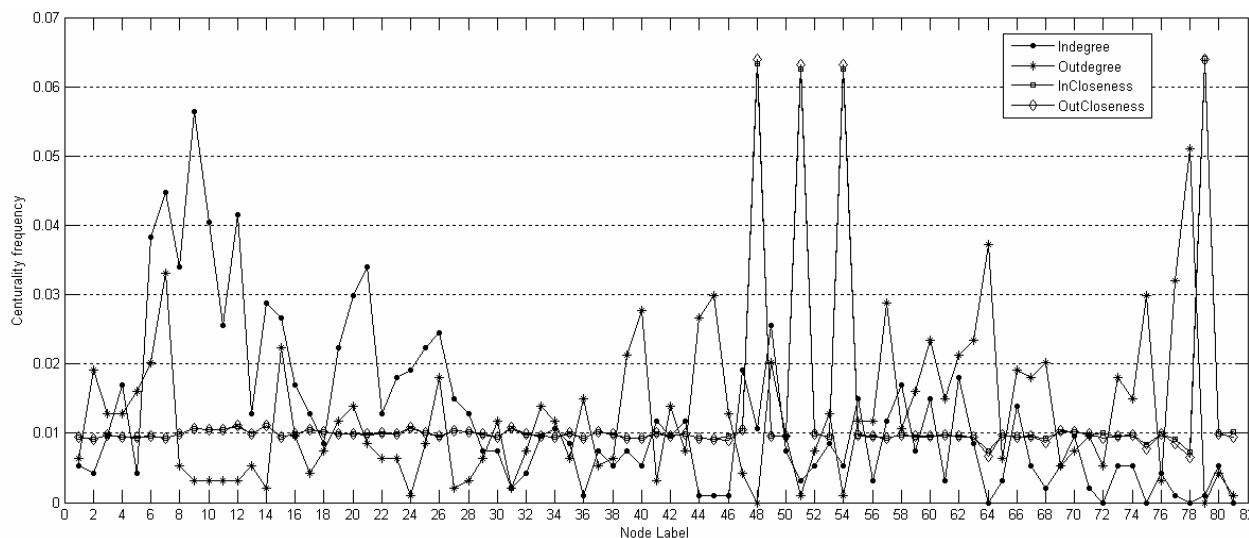


Figure 5. The centrality on electronic info. & computer communities

electronic information & computer community contains 88 nodes, and there are 19 nodes with indegrees bigger than 17, in which nearly 70% nodes with outdegrees smaller than 10. The journal with biggest indegree is “Computer Engineering and Applications”. Its indegree value is up to 59, but only has an outdegree value of 14. To be contrast, “Journal of Beijing University of Posts and Telecommunications” as the journal with biggest outdegree of 27, only having an indegree of 5. The degree distribution characteristic induces a strong unidirectional citation pattern in this kind of communities. On the other hand, though the incloseness and outcloseness of each node are approximately consistent, the whole picture shows sharp changes. Most nodes have small closeness, except four of them, which are “Computing Techniques for Geophysical and Geochemical Exploration”, “Robots”, “Piezoelectrics and Acoustooptics” and “Electronic Components & Materials”. These four nodes all locate near the edge of network, as shown in Figure 8. left. And their large closenesses

indicate the loose citation relations between them and other nodes.

In the same way, taking the light & textile industry community as an example for the second kind of community to analyze its centrality, and the tendency is shown in Figure 6. Nodes with large indegrees usually also have large outdegrees, and vice versa. The node with maximum in-degree and maximum out-degree all belongs to “Food Science”, the two values of which don’t have too much difference (in-degree=11, out-degree=7). This kind of degree distribution presents tight connections among the nodes in communities. With further consideration on closeness centrality, it is easy to figure out nodes in the light & textile industry distribute quite even from the globe network, because the closeness curve displays in a gentle way. Figure 7 gives a directly look on node closeness, where node sizes are consistent with their closeness value. This phenomenon tells a further illustration on the bidirectional citation tendency of the light & textile industry community.

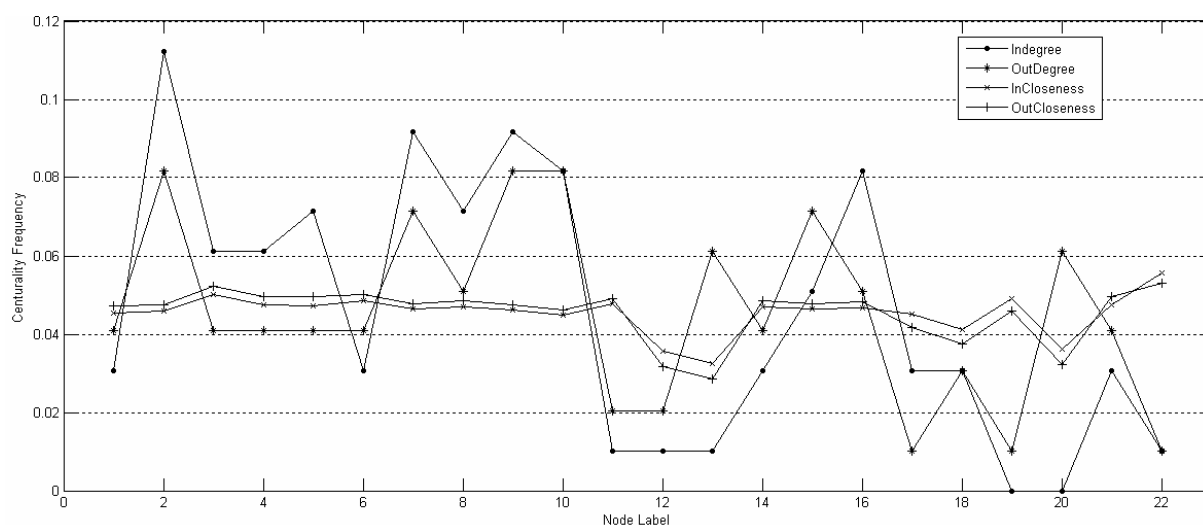


Figure 6. The centrality on light & textile industry communities

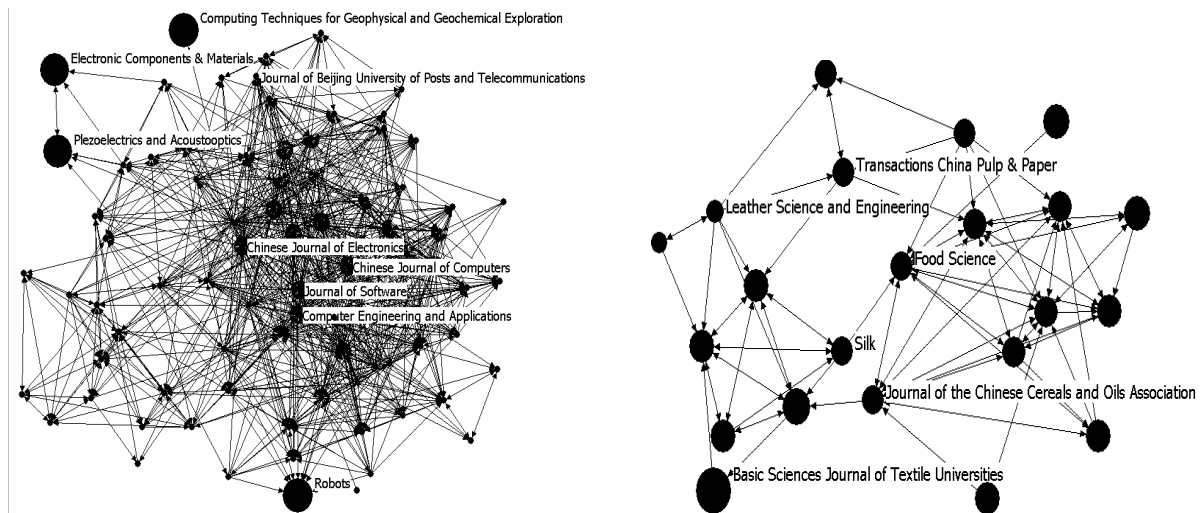


Figure 7. Centrality of journal network communities (left shows the electronic info. & computer community, right shows the light & textile industry community)

5. Conclusions

Chinese journal citation network is abstracted from more than one and a half thousands of Chinese journals of science and technique by CSTPC index. It is found these networks have obvious clustering characteristic and small-world pattern. This paper also borrows the motif concept into consideration to present some structure differences between two different kinds of network communities. One kind is more inclined to unidirectional citation pattern, while the other prefers the bidirectional citation ones. Then we give a further investigation on the reason of these two different kinds of citation patterns, according to node centrality in the communities. With a detailed statistics on node degree and its closeness, it illustrates communities of different kind also share different centrality characteristics.

Unlike general methods, this research takes three-node motifs as a basic granularity to find the discrepancy between different communities, rather than on a traditional node granularity. And it also gets some interesting ideas on journal citation networks. In the future, we could probably consider using motifs, a higher granularity to be a community partition criterion, instead of only using the system units, node or edge.

6. Acknowledgments

This work is partially supported by the National Grand Fundamental Research 973 Program of China under Grant No. 2007CB310803 and the National Natural Science Foundation of China under Grant No. 60675032.

REFERENCES

- [1] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, 393(6684): pp. 440–442, 1998.
- [2] A. L. Barabasi and R. Albert, "Emergence of scaling in random networks," *Science*, 286(5439): pp. 509–512, 1999.
- [3] R. Milo, S. Shen-Orr, et al., "Network motifs: Simple building blocks of complex networks," *Science*, 298: pp. 824–827, 2002.
- [4] T. I. Lee, et al., "Transcriptional regulatory networks in *Saccharomyces cerevisiae*," *Science*, 298: pp. 799–804, 2002.
- [5] D. T. Odom, et al., "Control of pancreas and liver gene expression by HNF transcription factors," *Science*, 303: pp. 1378–1381, 2004.
- [6] N. Iranfar, D. Fuller, and W. F. Loomis, "Transcriptional regulation of post-aggregation genes in *Dictyostelium* by a feed-forward loop involving GBF and LagC," *Developmental Biology*, 290: pp. 460–469, 2006.
- [7] R. Prill, P. Iglesias, and A. Levchenko, "Dynamic properties of network motifs contribute to biological network organization," *PLoS Biology*, 3: pp. e343, 2005.
- [8] E. Ravasz, A. L. Somera, D. A. Mongru, et al., "Hierarchical organization of modularity in metabolic networks," *Science*, 297: pp. 1551–1555, 2002.
- [9] R. Milo, S. Itzkovitz, N. Kashtan, et al., "Superfamilies of evolved and designed networks," *Science*, 303: pp. 1538–1542, 2004.
- [10] P. Zhou, L. Leydesdorff, and Y. S. Wu, "The visualization of Chinese Journal of Scientific and Technic in citation environment," <http://users.fmg.uva.nl/lleydesdorff/istic03/index.htm>.
- [11] G. Kossinets and D. J. Watts, "Empirical analysis of an evolving social network," *Science*, 331: pp. 88–90, 2006.
- [12] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, 99: pp. 7821–7826, 2002.
- [13] J. Tyler, D. Wilkison, B. Huberman, "Email as spectroscopy: Automated discovery of community structure within organizations," *International Conference on Communities and Technologies*, pp. 81–96, 2003.
- [14] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, "Defining and identifying communities in networks," *Proceedings of the National Academy of Sciences*, 101: pp. 2658–2663, 2004.
- [15] S. Fortunato, V. Latora, and M. Marchiori, "A method to find community structures based on information centrality," *Physical Review E*, 70: 056104, 2004.
- [16] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, 69: 066133, 2004.
- [17] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, 435 (7043): pp. 814–818, 2005.

Two-Tier GCT Based Approach for Attack Detection

Zhiwen Wang, Qin Xia, Ke Lu

MOE KLINNS Lab and SKLMS Lab, Department of Computer Science & Technology, Xi'an Jiaotong University, Xi'an, 710049, P.R.China

Email: wzw@mail.xjtu.edu.cn, qxia@ctec.xjtu.edu.cn, luke@stu.xjtu.edu.cn

Received November 23rd, 2008; revised November 27th, 2008; accepted December 2nd, 2008.

ABSTRACT

The frequent attacks on network infrastructure, using various forms of denial of service attacks, have led to an increased need for developing new techniques for analyzing network traffic. If efficient analysis tools were available, it could become possible to detect the attacks and to take action to weaken those attacks appropriately before they have had time to propagate across the network. In this paper, we propose an SNMP MIB oriented approach for detecting attacks, which is based on two-tier GCT by analyzing causal relationship between attacking variable at the attacker and abnormal variable at the target. According to the abnormal behavior at the target, GCT is executed initially to determine preliminary attacking variable, which has whole causality with abnormal variable in network behavior. Depending on behavior feature extracted from abnormal behavior, we can recognize attacking variable by using GCT again, which has local causality with abnormal variable in local behavior. Proactive detecting rules can be constructed with the causality between attacking variable and abnormal variable, which can be used to give alarms in network management system. The results of experiment showed that the approach with two-tier GCT was proved to detect attacks early, with which attack propagation could be slowed through early detection.

Keywords: Network Behavior, Attack Detection, Granger Causality Test, Management Information Base

1. Introduction

The frequent attacks on network infrastructure, using various forms of denial of service (DoS) attacks and worms, have led to an increased need for developing techniques for analyzing and monitoring network traffic. If efficient analysis tools were available, it could become possible to detect the attacks and take action to suppress them before they have had much time to propagate across the network. In this paper, we study the possibilities of SNMP MIB based mechanisms for attack detection.

Detecting attacks close to the attacker allows us to limit the potential damage close to the target. Traffic monitoring close to the source may enable the network operator quicker identification of potential attack and allow better control of administrative domain's resources. Attack propagation could be slowed through early detection.

The current approach passively monitors network traffic at regular intervals and analyzes it to find any abnormalities. By observing the traffic and correlating it to previous states of traffic, it may be possible to see whether the current traffic is behaving in a correct manner. The network traffic could be different because of flash crowds, changing access patterns, infrastructure problems such as router failures, and DoS attacks. In the case of bandwidth attacks, the usage of network may be

increased and abnormalities may show up in traffic volume. These approaches rely on analyzing regularity of network traffic in order to provide indications of possible attacks in the traffic.

However, all the approaches on detecting attack mentioned above absolutely depend on individual network behavior at the target, which usually ignore the causality among different network behaviors and the impact of time series. Those impacts may be caused by attacking behaviors at the attacker in most cases, so it is prone to produce a high rate of failed and false alarm [1]. It's important to study how to construct network behaviors influenced by attacks in a complex environment. The causal relationship of network behavior between the attacker and the target make it become possible to detect the attacks early at the attacker and to take appropriate action to weaken those attacks before they have had time to propagate across the network.

In this paper an SNMP MIB oriented approach based on two-tier GCT (Granger Causality Test) is presented, which can detect attack before the security was damaged at the target. According to the abnormal behavior constructed at the target, GCT is executed initially to find preliminary attacking variable, which has whole causality with abnormal variable in network behavior. Relying on the behavior features extracted from abnormal behavior, GCT is executed again to recognize attacking variable,

Funding for this work was provided by China NSF Grant (60633020, 60473136, 60373105), and National High Tech. Development Plan (2006BAH02A24-2, 2006BAK11B02, 2007AA01Z475).

which has local causality with abnormal variable in local behavior. The causality between attacking and abnormal variable is used to build detecting rules. These detecting rules make it possible to detect attacks at the attacker early. SNMP MIB traffic variable of *udpOutDatagrams* is successfully recognized as attacking variable and detecting rules was built well under the experiment of Trin00 UDP Flood. The final results showed that the approach with two-tier GCT is proved to detect attacks at the attacker early, which has great effect on slowing the attack propagation to the target.

This paper makes the following contributions: 1) considers the time series analysis of network behaviors; 2) presents a novel approach based on two-tier GCT for detecting attack; 3) uses prevalent SNMP MIB traffic variable as input of detecting model; and 4) shows the approach with two-tier GCT is more accurate than that with single GCT under the experiment of Trin00 UDP Flood.

The rest of the paper is organized as following. Section 2 gives an overview of related work. Section 3 analyses the time sequence of network attack. Section 4 gives some basic definitions and presents the correlation method and correlating procedure of network behavior. Section 5 describes a novel approach on detecting attack based on two-tier GCT, which is SNMP MIB traffic variable oriented. Trin00 UDP Flood experiment is carried out in Section 6, which shows the effect that attack propagation could be slowed through early detection. Section 7 draws conclusions of the paper.

2. Related Work

Many approaches have been studied to detect, prevent and mitigate malicious network traffic. For example, rule-based approaches, such as IDS, try to apply previously established rules against incoming traffic to detect and identify potential DoS attacks close to the victim's network. To cope with novel attacks, however, IDS tools such as Snort [2] require to be updated with the latest rules. This paper pays attention to the problem of designing generalized measurement based real-time detection mechanisms. Measurement-based studies have considered traffic volume [3,4,5], number of flows [6] as potential signals that can be analyzed in order to detect anomalies in network traffic, while we further utilize the SNMP MIB traffic variables such as *ipOutRequests*, *udpInDatagrams*, *tcpInErrs*, etc. Work in [5] relies on input data from multiple sources, while our work focuses on the traffic variables located in each machines.

Some approaches proactively seek methods to suppress the overflow of traffic at the source [7]. Controls based on rate limits have been adopted for reducing the monopolistic consumption of available bandwidth, to diminish the effects of attacks, either at the source or at the destination [7,8,9]. The apparent symptoms of bandwidth attack may be sensed through monitoring bit rates [10] and/or packet counts of the traffic flow.

Bandwidth accounting mechanisms have been suggested to identify and restrain attacks [11,12,13,14,15,16]. Packeteer [17] and others offer commercial products that can account traffic volume along multiple dimensions and allow policy-based rate control of bandwidth. Pushback mechanisms have been proposed to contain the detected attacks closer to the source [9,13,18]. Traceback has been proposed to trace the source of DDoS attacks even when the source addresses may be spoofed by the attacker [19]. Seong [20] proposes a traffic anomaly detector, operated in postmortem and in real-time, by passively monitoring packet headers of traffic.

However, sophisticated low-rate attacks [21], which do not give rise to noticeable variance in traffic volume, could go undetected when only traffic volume is considered. Recently statistical analysis of aggregate traffic data has been studied. In general, the generated signal can be analyzed by employing techniques such as FFT (Fast Fourier Transform) and wavelet transforms. FFT of traffic arrivals may reveal inherent flow level information through frequency analysis. Fourier transforms and wavelets have been applied to network traffic to study its periodicity [22,23].

Among the detecting methods, Cabrera first attempted to detect network attack by using GCT whose core is to check whether the lag information of a random variable will make an statistically effective forecasting to another random variable with statistical tools [24]. GCT has been applied to many fields successfully, such as earthquake warning, stock-market analyzing, network security etc. Cabrera carried out an experiment on detecting attack in which SNMP MIB was chosen to act as detecting variables in order to recognize some attacking variables reflecting the attacking procedure, but the time interval between units in the same data series is too long to reflect the causality between data series exactly. WANG Sheng [25] considered that attacking procedure may have various causality in whole and local network behavior, and he put forward the idea of GCT based on local data series. There is no experiment done by WANG to support his idea.

Based on the foundation mentioned above, the detecting method of Causality in network behavior was studied in-depth by making full use of existing SNMP MIB traffic variables. A novel approach with two-tier GCT characterized by *whole causality first*, *local causality second* is presented in this paper and will be described detailed in below sections.

3. Time Sequence of Attack

Typical network attack includes spatial and temporal dimensions. Spatial dimension means the physical location of network entities involved in attacking procedure is arbitrary, and temporal dimension means there is time sequence between mutual interactions produced by network entities involved in an attacking procedure. The time sequence of network attack is

depicted in Figure 1, where a complete attacking procedure consists of the following four stages.

1) Prepare to attack (T_0). Attacker scans vulnerabilities and identifies system to choose target.

2) Attacking (T_1). Attacker initiates attacking command, such as TCP semi-connection, ICMP Flood etc.

3) Attack takes effect (T_2). Attacking command arrives at target and leads to abnormal behavior on target.

4) Target damaged (T_3). Sustained attacks make the security of target damaged.

The arbitrary of spatial distribution and uncertain of time lag exacerbate the complexity of detecting attack. The common principle of detecting approaches is that relevant data originating from temporal dimension or spatial dimension is collected first, and then some methods, such as rules reasoning, FSM, pattern matching and statistical analysis are applied to extract the feather of network attack so as to avoid the attacking procedure to enter in T_3 or T_2 stage.

4. Behavior Correlation Method

4.1 Definition

In order to describe the approach with two-tier GCT used for detecting network attack exactly, some necessary items are defined as follows.

1) Network behavior. The numerical value sequence of detecting variables which represents the running state of network, such as CPU utilization, available network bandwidth and memory consumption, is observed over a continuous period and which is denoted by $B=\{v_k\}$ ($k=1,2,3,\dots,N$), where v_1 and v_N stand for the value of detecting variable V at the starting time t_{init} and end time t_{end} respectively. The variable $t_{interval}=(t_{init} - t_{end})/N$ is defined as observation interval, which will directly affect the accuracy of network behavior description.

2) Time window. The part of detecting time corresponding to constructing the network behavior, denoted by $W(t_{low}, t_{upper})$, where t_{low} and t_{upper} stand for the bottom and top of a time window respectively. The difference of top and bottom is defined as time window size t_{win} .

3) Behavior feature. Some certain regularity in a time window or among time windows is showed by the observational numerical value in network behavior. The behavior feature is denoted as $F=\{v_\lambda\} \subseteq B$ ($\lambda=1,2,3,\dots,n$), where v_1 and v_n stand for the observational numerical values corresponding the time of t_{low} and t_{upper} . There are five types of regularity for behavior feather, ie. ① The observational numerical value is increased monotonously during a time window. ② The observational numerical

value is diminished monotonously during a time window.

③ The observational numerical value is above the special threshold in a time window. ④ The observational numerical value is below the special threshold in a time window. ⑤ The observational numerical value is changed in periodicity among time windows.

4) Local behavior. Defined as the observation numerical sequence which is acquired when t_{low} of a time window corresponding to behavior feature is moved backward a time window size, and whose length is double of the length of behavior feature on the time sequence.

5) Abnormal behavior. The network behavior represented by network entities on targets whose security will be damaged at stage of T_3 or T_2 . The detecting variables used in constructing abnormal behaviors are called abnormal variables.

6) Attacking behavior. The network behavior represented by attacker at stage of T_0 or T_1 , which will damage the security of one or more network entities with some possibility.

7) Preliminary attacking variables. The detecting variables which has whole causality with abnormal variables in network behavior.

8) Attacking variables. The detecting variables which has local causality with abnormal variables in local behavior. Attacking variables are always used in constructing attacking behaviors.

9) Behaviors correlating. The procedure which is to mine the causality between abnormal variables and detecting variables with GCT. There are two types of behaviors correlation named whole correlation and local correlation respectively. The former is used to find preliminary attacking variables and the later is used to recognize attacking variables.

10) Detecting Rule. The reflection of causality between attacking variable and abnormal variable, denoted as $(\{V_{attack}\}, V_{abnorm})$, which make attacker oriented detection possible.

4.2 Correlation Method

Given a large database describing the operation of an Information System, we view the problem of extracting proactive Detecting Rules for security as consisting of the three steps delineated below. These steps are performed off-line, and produce a set of rules to be used for detecting security violations on-line. The correlation of causal relationship can be inferred from measured variables in this paper.

1) Detecting Anomaly. The objective here is to determine the variable in the target machine, which is better characterizing the occurrence of an attack. The final product of this step is the list of abnormal variables at the target. There are two procedures for determining the abnormal variable at the target. One way is to use domain knowledge about the special attack. For example, for Ping Flood, it is known that *icmpInEchos* is the right

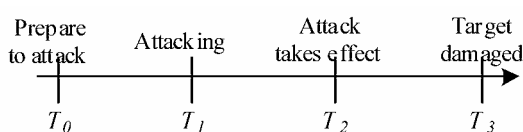


Figure 1. Time sequence of network attack

variable to look for, since Ping Floods are affected by sending much *icmpInEchos* packets to a target. A second way is to compare the evolution of each variable during an attack with the evolution of the variable during normal operation. Variables that display a large variation between normal operation and attack should be declared abnormal variables at the target. Since we are looking for localized variation in the variable, the time series should be segmented on small sub-time series, which are then compared with normal profiles. This procedure was used in [26] for detecting anomalies in network operation due to component faults. Anomalies were detected as variation on the parameter of AutoRegressive models. In this paper, we will utilize domain knowledge about the attacks for extracting the abnormal variable at the target.

2) Computing Correlation. Once the abnormal variable at the target are determined, we need to determine variables in the prospective attacker that are causally related with them. These variables at the attacker are related to T_2 and T_3 events. Recall that we do not know which ones are the attacker. We only know a list of candidates and their corresponding variables. We make the assumption that any causal relationship between variables at prospective attackers and the abnormal variables at the target is to be inferred as a link between that attacker and the target. The final product of this step is the list of attacking variables.

3) Constructing Detecting Rules. Following the computing correlation, the objective here is to extract particular features of the attacking variables at the attacker that precede the attack at the target. Recall that these variables were found to be causally related with the attack; hence we may expect that certain anomalies in these variables can be indicative of an incoming attack. Once these features are determined and are shown to precede the attack, we can construct proactive detecting rules that constitute the end product of this step. These rules can be used to implement alarms on a network management system.

4.3 Network Behavior Correlation

According to above definition we attempt to recognize the variables at the attacker that are causally related to the abnormal variables at the target. Since we are looking for proactive detecting rules, we should recognize variables at attacker which contains events that precede the damage at the target. These events can be T_2 events, or T_3 events, as described in Section 3. In this section, the use of Causality Tests is to be investigated for correlating the network behaviors at the attacker with the network behavior at the target [27]. Testing for causality in the sense of Granger, involves using statistical tools for testing whether lagged information on a variable u provides any statistically significant information about another variable y . if not, then u does not Granger-cause y . GCT compares the residuals of an AutoRegressive Model with the residuals of an AutoRegressive Moving Average Model. Assuming a particular lag length P , and estimate the following unrestricted equation.

$$y(k) = \sum_{i=1}^P \alpha_i y(k-i) + \sum_{i=1}^P \beta_i u(k-i) + e_1(k); \quad k = 0, 1, 2, \dots, N-1$$

The null hypothesis of the H_0 GCT is given by:

$$H_0: \beta_i = 0, \quad i = 1, 2, \dots, p$$

i. e. u does not affect y up to a delay of p units. The null hypothesis is tested by estimating the parameters of the following restricted equations.

$$y(k) = \sum_{i=1}^P \gamma_i y(k-i) + e_0(k)$$

Let R_1 and R_0 denote the sum of the squared residuals under the two cases.

$$R_1 = \sum_{t=1}^T e_1^2(t), \quad R_0 = \sum_{t=1}^T e_0^2(t); \quad T = N - P$$

If the test statistic g given by

$$g = \frac{(R_0 - R_1)/P}{R_1/(T - 2P - 1)} \sim F(P, T - 2P - 1)$$

is greater than the specified critical value, then reject the null hypothesis that u does not Granger-cause y . Here, $F(a, b)$ is Fisher's F distribution with parameter a and b . In other words, high values of g are to be understood as representing strong evidence that u is causally related to y . In the traditional sense, we say that u_1 is more likely to u_2 to be causally related with y if $g_1 > g_2$, where g_i , $i=1, 2$ denote the GCT statistic for the input-output pair (u_i, y) .

5. Our Approach

Our approach, which is based on two-tier GCT, is modeled in Figure 2. The model consists of four main components, ie constructing network behavior, detecting anomaly, recognizing attacking variable and preventing attack.

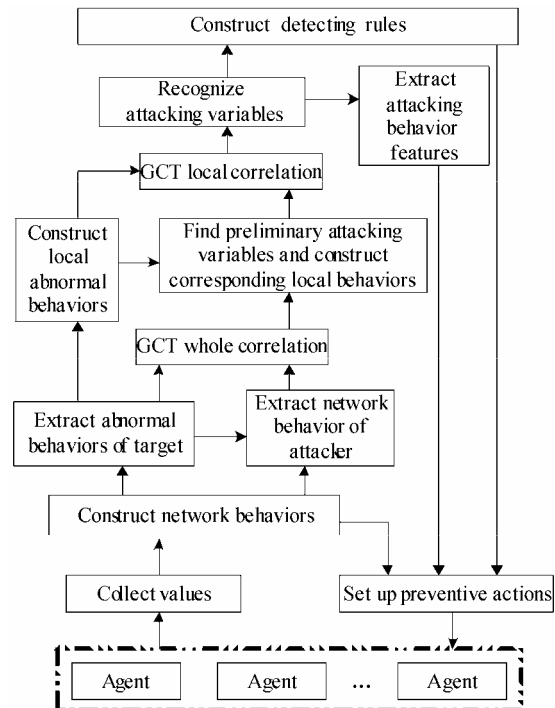


Figure 2. Two-tier GCT based approach

5.1 Constructing Network Behavior

According to the definition of network behavior, we find that SNMP/MIB is able to meet the requirements of detecting variables completely. There are still three technical problems needed to be solved before the appropriate network behavior is constructed for detecting attack.

1) Choose detecting variables. In order to reduce the number of network behavior and improve the accuracy of recognizing detecting variables, it is necessary to choose detecting variables from SNMP/MIB exactly. As we know, both attacker and target act as network termination entities in most cases, and the data transmission between them is executed on network layer or higher layer, so we are like to choose 32 variables from IP, ICMP, TCP and UDP variable group as detecting variables, which represent the dynamic performance of network.

2) Decide the way of collecting value. Collecting values of detecting variables period is a necessary step for constructing network behaviors correctly. Various ways will bring different effect in collecting values and polling is rather appropriate because of its simplicity and robust.

3) Determine the period of network behavior. According to working situation network behaviors can be measured by hour, day, week or month. The fact that normal network traffic is varying in a one-day circle is found in reference [28], which is accomplished through many times of observation and experiment. The period of network behavior is measured by day in this paper.

5.2 Detecting Anomaly

The key of detecting anomaly is to recognize abnormal variable from detecting variables. There are two methods used commonly, which include special analysis based on domain knowledge and statistical analysis. The former is suitable to attacks with manifest procedure, by which abnormal variable can be recognized directly from SNMP MIB by making use of domain knowledge. The latter is more suitable to attacks where abnormal variables can't be recognized directly through the attack procedure. Statistical deviation of network behavior must be calculated for every detecting variable in attacking and normal environments independently, and detecting variables with the largest deviation are confirmed to be abnormal variables.

After abnormal variables were recognized, attack test can be carried out repeatedly and abnormal behavior will be monitored successively. As a result, the abnormal behavior feature can be extracted by checking whether it is consistent with the behavior feature defined in Section 4.

5.3 Recognizing Attacking Variables

Causality in network behavior between attacker and target can be correlated based on the features of attack procedure with time backward tracking. According to the

abnormal behavior on target, preliminary attacking variables which have whole causality with abnormal variables in network behavior can be found first from detecting variables by using GCT. Then according to the behavior features of abnormal behaviors, attacking variables which has local causality with abnormal variables in local behaviors can be recognized from preliminary attacking variables by using GCT again. The detecting variables whose value exceeds the threshold set in the two GCT will be recognized as attacking variables.

The whole correlation in network behavior is processed as following.

1) Obtain abnormal behavior b_{abnorm} from network behavior base on target,

2) Obtain all the network behaviors $h_{attack}(j)$ from network behavior base on attacker, which are coincided with b_{abnorm} in detecting period;

3) Calculate the GCT detection statistics g_{whole} of all input/output pair $h_{attack}(j), b_{abnorm}$;

4) If g_{whole} corresponding to any $h_{attack}(j)$ is beyond the critical value F_{α} of F distribution under significance level α , it is showed that $h_{attack}(j)$ has whole causality with b_{abnorm} and the detecting variables used to construct $h_{attack}(j)$ will be recognized as preliminary attacking variables [27].

Because GCT is a statistical method, the preliminary attacking variables recognized by executing whole correlation only once is some fortuitousness. As a result, it is necessary to execute whole correlation many times so as to recognize preliminary attacking variables with more accuracy.

The process of local correlation between local behaviors is described as follows.

1) Extract all behavior features of abnormal behavior b_{abnorm} , denoted as $f(i), i=1,2,3,...,M$;

2) Construct local abnormal behavior corresponding to the abnormal behavior feature, denoted as $local_b_{abnorm}(i), i=1,2,3,...,M$.

3) If $h_{attack}(j)$ has whole causality with b_{abnorm} , local behavior $local_h_{attack}(i, j)$ which is in the same detecting period with $local_b_{abnorm}(i)$ will be constructed.

4) Calculate GCT statistics $g_{local}(i, j)$ of all input/output data pairs $(local_h_{attack}(i, j), local_b_{abnorm}(i))$.

5) If $g_{local}(i, j)$ is below the critical value F_{α} of F distribution under significance level α , it's showed that $h_{attack}(j)$ doesn't have local causality with b_{abnorm} .

6) Define $g_{local}(j)$ of $h_{attack}(j)$ as the sum of $g_{local}(i, j)$ belong to the same $h_{attack}(j)$. The higher $g_{local}(j)$ is, the more possibility $h_{attack}(j)$ is recognized as attacking behavior.

$$g_{local}(j) = \sum_{i=1}^M (g_{local}(i, j) t_{win}(i)) / \left(\sum_{i=1}^M t_{win}(i) \right)$$

In the expression depicted above, $t_{win}(i)$ represents the size of time window corresponding to the i th behavior feature of abnormal behavior.

1) Construct the attack detecting rules according to the recognized attacking variables.

5.4 Preventing Attack

The attacking variables recognized at the attacker are labeled as causally related with the abnormal variables at the target, but we still need to find trigger, or a key event at the attacker. This is an anomaly detection problem. We postulate that any anomalous behaviors in attacking variables at the attacker are to be considered key events at the attacker. One possible approach is to look for jumps in the attacking variables, by monitoring the absolute values of the differentiated time series. Using many normal runs, we constructed a normal profile of jumps for each of the 32 MIB traffic variables. Given an attacking variable, key events at the attacker are defined as jumps larger than the largest jump encountered the normal profile of jumps. Those key events are used to set the alarms.

6. Experiment Simulation

The certainty of attacking variable and effect of attack detection will be verified in the following experiments in order to validate the approach with two-tier GCT.

Experimental environment consists of an attacker host, a target host and a security management host, which are connected through Ethernet. SNMP Agent is deployed on the attacker host and target host and the security management host is responsible for detecting attack. Trin00 UDP Flood [29] is selected on attacker in experiments. According to its principle, SNMP MIB traffic variable of *udpInDatagrams* is selected as abnormal variable in Trin00 UDP Flood. The unit of time for experiment is measured by days and the duration of each attack procedure persists for 1 hour. The value of 32 traffic variables acted as detecting variables at the attacker and *udpInDatagrams* at the target are collected every 10 seconds and 1 minute respectively. All tests are carried out against attacker under three types of running configuration, which is depicted as following.

- ① execute attack only
- ② execute attack and FTP
- ③ execute both attack and Netflow

6.1 Certainty of Attacking Variable

Based on detecting variable and abnormal variable, certainty of attacking variables is validated by checking whether the attacking variables recognized in different environments are identical. The results acquired by using single GCT (proposed by CABRERA in [24]) and two-tier GCT respectively are compared to validate the advantage of the approach presented in this paper.

Table 1 shows the critical value $F_{\alpha}(p, T-2p-1)$ of F distribution for GCT causality statistics g_{whole} and g_{local} under significance level α of 0.05. The approach with single GCT needs only whole correlation which computes the whole causality statistics g_{whole} in network behavior between each of 32 detecting variables and *udpInDatagrams* at the target. Among the detecting variables exceeding critical value F_{α} , one with the largest

g_{whole} is recognized as attacking variable. Table 2 shows the results of test with sampling interval of 1 minute.

It's different from the approach with single GCT, detecting variables with g_{whole} over critical value F_{α} are just treated as preliminary attacking variables in the approach with two-tier GCT. Comparing to the original 32 detecting variables, the number of preliminary attacking variables is reduced greatly, which is good to perform local correlation in local behavior and to reduce the cost of implementing GCT. Three monotonous increasing behavior features corresponding to the three attacking actions taken by attacker host are observed by analyzing the abnormal behaviors, and the duration of each is not the same as the duration of attacking action. In order to keep the consistency of detecting in time dimension, only the first 60 minutes of monotonous increase duration is considered as time window of behavior feature. Accordingly, the period of local behavior should be set by 120 minutes. In order to recognize attacking variable, each of the local causality statistics g_{local} between preliminary attacking variable and abnormal variable should be computed. The variable exceeding the critical value F_{α} with the largest g_{local} is recognized as attacking variable. Table 3 shows the attacking variable recognized by using two-tier GCT with sampling interval of 10 minutes.

By comparing the results in Table 2 and Table 3 we found that the attacking variable recognized with single GCT is uncertain in different environments, where *ipOutRequests* was recognized as attacking variable in the first 2 running configurations and *udpOutDatagrams* was recognized in the third running configuration. On the contrary, the attacking variable recognized with two-tier GCT is certain well, where *udpOutDatagrams* was recognized as attacking variable in three different running configurations.

6.2 Effect on Attack Detection

To demonstrate the effect of attack detection with attacking variables *ipOutRequests* and *udpOutDatagrams* independently in preventing attack, an experiment lasted 5 days was carried out incessantly. Trin00 UDP Flood was initiated random by 10 times for each day in the identical running environments configured as before. The detecting results acquired with *udpOutDatagrams* and *ipOutRequests* respectively were listed in Table 4. According to the results, we found that the success rate of detection with *udpOutDatagrams* is significantly higher than detection with *ipOutRequests*. It's obvious that the performance of approach with two-tier GCT is better than the approach with single GCT.

Table 1. Critical value of F distribution

statistics	interval	times	P	T	95%
g_{whole}	1 min	1440	200	1240	1.19
g_{local}	10 s	720	100	620	1.28

Table 2. Results acquired with single GCT

running configuration	number of detecting variable	maximum of g_{whole}	minimum of g_{whole}	number of detecting variable satisfying $g_{whole} \geq F_a$	attacking variable
①	32	4.11	1.04	8	ipOutRequests
②	32	3.67	0.91	7	ipOutRequests
③	32	3.50	0.79	11	udpOutDatagrams

Table 3. Results acquired with two-tier GCT

running configuration	number of preliminary attacking variable	duration of abnormal features	maximum of g_{local}	minimum of g_{local}	number of detecting variable satisfying $g_{local} \geq F_a$	attacking variable
①	8	61.2	3.65	1.22	5	udpOutDatagrams
②	7	59.4	3.41	1.02	6	udpOutDatagrams
③	11	64.4	2.87	0.98	5	udpOutDatagrams

Table 4. Detecting effect with different attacking variables

running configuration	number of attacking variable	detecting with <u>udpOutDatagram</u>			detecting with <u>ipOutRequestss</u>		
		actual	false	failed	actual	false	failed
①	50	51	1	0	58	8	0
②	50	52	2	0	62	14	2
③	50	55	7	2	65	21	6

7. Conclusions

Since the conventional method of network attack detection is focused on stage T_3 of attacking procedure, it is difficult to detect the attack before security of target is damaged. An SNMP MIB oriented approach based on causality of network behavior is presented in this paper. According to the abnormal behavior features hidden in detecting variables on target in attacking procedure, backward retrospection is executed twice with two-tier GCT. Depending on whole causality between detecting variables and abnormal variables the preliminary attacking variables is found first. Then according to behavior features extracted from abnormal behaviors, attacking variables which has local causality with abnormal variables can be recognized by using GCT again and the corresponding rules for attack detecting can be constructed subsequently. The results of experiment showed that the approach was proved to detect attack on attacker, which has effect on blocking the pervasion of attacking procedure to target.

As an on-line detecting method, the approach with two-tier GCT employs small amounts of SNMP MIB traffic data in order to keep such analysis simple and efficient. At the same time, the data cannot be so small that meaningful statistical conclusions cannot be drawn. However, on-line detection may also require that any indications of attacks be provided with short latencies. The tension between robustness and latency makes on-line detection more challenging.

REFERENCES

- [1] M. Thottan and C. Y. Ji, "Anomaly detection in IP networks," IEEE Transactions on Signal Processing, 51(8): pp. 2191–2204, 2003.
- [2] M. Roesch, "Snort-lightweight intrusion detection for networks," in USENIX LISA 1999, Seattle, WA, November 1999.
- [3] P. Barford et al., "A signal analysis of network traffic anomalies," in ACM SIGCOMM Internet Measurement Workshop, November 2002.
- [4] A. Hussein, J. Heidemann, and C. Papadopoulos, "A framework for classifying denial of service attacks," in ACM SIGCOMM, August 2003.
- [5] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," in ACM SIGCOMM, September 2004.
- [6] D. Plonka, "FlowScan: A network traffic flow reporting and visualization tool," in USENIX LISA 2000, New Orleans, LA, December 2000.
- [7] J. Mirkovic, G. Prier, and P. Reiher, "Attacking DDoS at the source," in IEEE International Conference on Network Protocols, November 2002.
- [8] A. Garg and A. L. N. Reddy, "Mitigation of DoS attacks through QoS regulation," in Proceedings of IWQOS, May 2002.
- [9] J. Ioannidis and S. M. Bellovin, "Implementing pushback: Router-based defense against DDoS attacks," in Proceedings of Network and Distributed System Security Symposium, February 2002.
- [10] Y. Zhang, L. Breslau, V. Paxson, and S. Shenker, "On the characteristics and origins of internet flow rates," in ACM SIGCOMM, August 2002.
- [11] Smitha, I. Kim, and A. L. N. Reddy, "Identifying long term high rate flows at a router," in Proceedings of High Performance Computing, December 2001.
- [12] I. Kim, "Analyzing network traces to identify long-term high rate flows," M. S. thesis, TAMU-ECE-2001-02, May

- 2001.
- [13] R. Mahajan, et al., "Controlling high bandwidth aggregates in the network," *ACM Computer Communication Review*, Vol. 32, No. 3, July 2002.
 - [14] C. Estan and G. Varghese, "New directions in traffic measurement and accounting," in *ACM SIGCOMM*, August 2002.
 - [15] A. Medina et al., "Traffic matrix estimation: Existing techniques and new directions," in *ACM SIGCOMM*, August 2002.
 - [16] D. Tong and A. L. N. Reddy, "QOS enhancement with partial state," in *Proceedings of IWQOS*, June 1999.
 - [17] Packeteer, "PacketShaper Express," white paper, 2003, http://www.packeteer.Com/resources/prod-sol/Xpress_Whitepaper.pdf.
 - [18] S. Floyd, S. Bellovin, J. Ioannidis, K. Kompella, R. Mahajan, and V. Paxson, "Pushback messages for controlling aggregates in the network," *IETF Internet draft*, work in progress, July 2001.
 - [19] S. Savage, D. Whetherall, A. Karlin, and T. Anderson, "Practical network support for IP traceback," in *ACM SIGCOMM*, 2000.
 - [20] S. S. Kim and A. L. N. Reddy, "Statistical techniques for detecting traffic anomalies through packet header data," *IEEE/ACM Transaction on Networking*, Vol. 16, No. 3, pp. 562–575, June 2008.
 - [21] A. Kuzmanovic and E. Knightly, "Low-rate TCP-targeted denial of service attacks," in *ACM SIGCOMM*, Karlsruhe, Germany, August 2003.
 - [22] A. Feldmann, A. Gilbert, P. Huang, and W. Willinger, "Dynamics of IP traffic: A study of the role of variability and the impact of control," *ACM Computer Communication Review*, Vol. 29, No. 4, pp. 301–313, 1999.
 - [23] C. M. Cheng, H. T. Kung, and K. S. Tan, "Use of spectral analysis in defense against DoS attacks," in *IEEE Globecom*, 2002.
 - [24] J. B. D. Cabrera, L. Lewis, and X. Z. Qin, "Proactive detection of distributed denial of service attacks using MIB traffic variables—a feasibility study," *IEEE Transactions on Signal Processing*, 49(6): pp. 609–622, 2001.
 - [25] S. Wang, L. C. Sun, and G. Z. Gan, "Application research based on Granger causality test for attack detection," *Computer Applications*, 25 (6): pp. 1282–1285, 2005.
 - [26] F. Zhang and J. Hellerstein, "An approach to on-line predictive detection," in *proceedings of the Eighth International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems*, San Francisco, CA, IEEE Computer Society, pp. 549–556 August 2000.
 - [27] J. Hamilton, "Time series analysis," Princeton University Press, 1994.
 - [28] B. X. Zou and Z. Q. Yao, "A method to stabilize network traffic," *Journal of China Institute of Communications*, 25(8): pp. 14–23, 2004.
 - [29] P. J. Criscuolo, "Distribution denial of service — trin00, tribe flood network, tribe flood network 2000, and stacheldraht," CIAC–2319, Department of Energy — CIAC (Computer Incident Advisory Capacity), 2000.

Towards Automatic Transformation from UML Model to FSM Model for Web Applications

Xi Wang, Huaikou Miao, Liang Guo

School of Computer Engineering and Science, Shanghai University, Shanghai, 200072, China

Email: {w_whitecn, hkmiao, glory}@shu.edu.cn

Received November 17th, 2008; revised November 26th, 2008; accepted November 30th, 2008.

ABSTRACT

The need for automatic testing of large-scale web applications suggests the use of model-based testing technology. Among various modeling languages, UML is widely spread and used for its simplicity, understandability and ease of use. But rigorous analysis for UML model is difficult due to its lack of precise semantics. On the other hand, as a formal notation, FSM provides an avenue for automatic generation of test cases, but the requirement for mathematical basis makes itself academic inventions divorced from real applications. This paper proposes an approach to transforming UML model to FSM model, taking advantage of both languages. As our work focuses on the transformation of UML state diagrams to FSM models, a specific transformation mechanism is presented, which deals with different elements with different mapping rules. To illustrate the mechanism we proposed, an example of a web application for software download is presented. Finally, we give a method for implementation of the mechanism and a tool prototype to support the method.

Keywords: UML Model, FSM Model, Model transformation

1. Introduction

Providing greater assurance that the software is of high quality and reliability, testing has been considered more and more important as people gradually realize the great effect on their daily life made by software products. Hand-crafted methods are acceptable until the coming of age when there are full of large-scale manufactures with high complexity, especially the appearance of web applications which labeled for their additional heterogeneity, concurrency and distribution.

Web applications are usually composed of front-end user interfaces, back-end servers including web servers, application servers and database servers, which build up a new way for deploying software applications. Components called for supporting task completion of web applications by each server may be programmed in different languages and executed on different platforms. In addition, web applications are frequently modified due to continuous updates of its components, high-speed developing technologies and changes of the needs of its users. All of these characteristics are challenging the traditional testing method which largely depends on the testers. On the other hand, most companies keep the minimum amount of time as their primary priority to meet market demand while customers pay their much attention to the reduction of the cost during maintenance, leading directly to the calls for effective testing within a relative short period of time.

Generation of test cases is the main task of testing; since detections of faults are operated by comparing expect outputs with actual ones obtained from running of these test cases. Model-based testing, which involves developing and using a model describing the structural and behavioral aspects of the system to generate test cases automatically, is an effective method for testing various software artifacts including web applications. As the models are developed early in the cycle from requirements information [1], the generation of test cases can be conducted in parallel with the implementation of the System Under Test (SUT), rather than sequentially, saving the time supposed to be spent for waiting. Also, it supports re-use in future testing as these models capture the behavior of a software system and in contrast to a test suite, they are much easier to update if the specification changes [2].

The critical part of model-based testing is the construction of models. Among various modeling languages, UML has been widely spread and used in industry for its simplicity and ease of use. It enables modelers to address all the views needed to analyze and develop the corresponding system. Further more, as a visual language, it can clearly show the structure and functions of the system, facilitating understanding and communication between designers, modelers, developers

and users. Besides, many powerful tools have been developed and used to support UML modeling such as argoUML. But unfortunately, it is widely acknowledged that UML can hardly provide formal semantics, as it comprises several different notations with no formal semantics attached to the individual diagrams. Therefore, it is not possible to apply rigorous automated analysis or to execute a UML model in order to test its behavior, short of writing code and performing exhaustive testing [3].

As one of the formal notations, FSM (Finite State Machine) provides a significant opportunity for testing because it precisely describes what functions the software is supposed to provide in a form that can easily be manipulated by automated means [4]. Being applied to the testing process, its relative theory could be helpful and supported for enhancing efficiency. Furthermore, in addition to traditional software, a web application's behavior could also be modeled using FSMs theoretically and then test cases could be automatically generated by traversing the path through the FSM model of the application, with each distinct path comprising a single test case [5]. Besides, FSM model can be visualized to tell intuitively the direction to which a test case is going, since state-based specification languages are fairly easy to translate into a specification graph as they have natural graph representations [4]. Last, the transformation to FSM facilitates model checking which verifies certain property of the model. However, its requirement for mathematical basis limits the range of utilization.

This paper proposes a method for transformation from UML model to FSM model, taking advantage of both: the simplicity and intelligibility of UML and the accuracy and derivability of FSM. It also enables the reuse of the existing and well-established tools for UML and theories for FSM. There're several kinds of diagrams within UML corresponding to different views of the system, our job focuses on the transformation of state diagram, as it is most often used to model the behavior of an individual object.

The remainder of this article is organized as follows: Section 2 reviews existing works in transformation of UML models. Section 3 presents a transformation mechanism from UML state diagrams to FSM models. To illustrate the transformation mechanism, an example of transforming from a state diagram representing a web application for software download is given in Section 4. In section 5, a method for implementing the transformation mechanism we proposed is given, together with a brief introduction to a tool prototype based on this method. Finally, concluding remarks and discussions about future works are presented in section 6.

2. Related Works

Automatic testing has become a hot spot in the software engineering field for facilitating development process of

software products. But most of the current technologies are based on "capture/replay" mechanism, which costs too much time and manual works while recording testing scenarios and handling with small changes on the functional design or user interfaces. Tools running on this mechanism will not design or generate test cases themselves and will not provide any instruction on the coverage situation of the generated test cases. Further more, there are even fewer automatic testing tools for web applications which requires for even more automatism. Most of the present tools [13] do not support the function test of web applications including Link Checks for checking links of the web application, HTML Validators for providing standard HTML syntax validation, Web Functional/Regression Test Tools, Web Site Security Test Tools, Load and Performance Test Tools and etc. Since most of them rely on information obtained from codes of the web applications and only concentrate on verification of static aspects, we need a tool to help verifying the behavior of them while paying least price.

With the appearance and popularity of the concept of object-oriented and model-driven, model-based testing for software products has aroused much attention in industry. Though many researches are done in this field, tools developed under their theories still have certain gaps with applying to real uses due to their lack of systematism and low automatic level [14,15,16,17,18,19].

Construction of models is the beginning of model-based testing for web applications. The most common one is to use Entity Relation Diagrams or UML Class Diagrams to model web pages of a web application and relationships between them. Isakowitz et al describe web applications with a method called Relationship Management Methodology [20]. Coda et al proposes a model WOOM for modeling web applications in a higher level of abstraction [21]. Gellersen et al introduce the WebComposition Markup Language for implementing a model for Web application development called Web Composition [22]. Conallen et al extend UML modeling language to model the structure of web applications [23]. However, these methods rarely construct models on the behavioral and functional aspects of the web applications and few testing approaches are figured out for these models.

The model language we use when designing the web applications is UML which strongly supports users to describe complicated software including web applications. But till now, no such complete testing tool has ever been implemented as its semi-formal semantics prevents it from automatic testing. On the other hand, many methods for generation of test cases from formal models are presented. [24] generates test cases from an Object-Oriented Web Test Model which is a combination of Object Relation Diagram, Page Navigation Diagram, Object State Diagram, Block Branch Diagram and Function Cluster Diagrams, but it will be trapped if there are too many objects in the software. Ricca et al models

web applications by modeling for each web page and obtain test cases according to proposed rules. Still, it would only be useful dealing with simple applications [5,25], models web applications with FSM which will then be used for test cases generation by search for different path of the model under different criteria. Considering that FSM model is also the most common used object for model checking, we choose it for destination of our model transformation process and origin for test cases generation.

Formalization of UML models has aroused much attention in industry. One of the most active group is the precise UML group [6], which is made up of international researchers who are interested in providing a precise and well-defined semantics for UML, by using model-oriented notations, such as Z or VDM. There are also works done by other researchers, Borges et al. [7] integrate UML class diagrams and a formal specification language OhCircus by written UML elements in terms of OhCircus. Latella et al. [8] converts UML state diagrams into the formal language Promela. Traore et al. [9] proposes a transformation mechanism from UML state diagrams to PVS which facilitates automatic model checking.

However, few researches on the transformation to the FSM model can be found. Erich et al. [11] gives a hierarchical finite state machine model for state diagrams, which is capable of acquiring the hierarchical information, but it does not mention the method for transformation to FSM models with the removal of hierarchy. [10] transforms time-extended UML state diagram into timed automata, but special elements of the state diagram are not under its consideration.

The method we proposed enables the transformation of state diagrams with special elements, such as *completion transition*, *fork*, *join* and *history state*. Besides, the flatness of the resulting FSM model can greatly support the automation of the generation of test cases.

3. Transformation Mechanism from UML State Diagram to FSM Model

As UML and FSM are source and target models of the transformation mechanism respectively, a brief introduction of both is given below.

3.1 UML

The Unified Modeling Language (UML) is becoming a standard language for specifying, constructing and documenting the artifacts of a software-intensive system. It can model from different perspectives with several kinds of diagrams that express static and dynamic aspects of a system. As a visualized model, UML conveys information intuitively to our human beings who can get better understanding through graphics. Besides, it is easy to learn and use, making it more attractive to those who model. Because of the characteristics mentioned above UML serves as the ideal model for describing the real.

Class diagram, object diagram, use case diagram, sequence diagram, communication diagram, activity diagram and state diagram are the most commonly used diagrams in UML. Class and object diagrams model the static design view of a system, mostly about relationships between objects, while rest of them focus on dynamic aspects. For the purpose of capturing unexpected outputs, we obtain most of the information needed for testing from behavioral models.

As one of the behavioral models, state diagram is often used to model the life cycle of certain object, from its motivation to termination. Since most systems involve more than one object, state diagrams are considered to be the minimal unit for representing behaviors. We therefore begin our research with UML state diagrams.

3.2 UML State Diagram

State diagram, which has been mainly discussed in this paper, specifies the sequences of situations an object goes through during its lifetime in response to events, together with its responses to those events. Many elements are involved for expressing semantics of the diagram.

States represent certain situations the object stays, each with a name for distinguishing itself from others. There are several types of states within state diagrams.

States that have no substructures are called simple states, others are called composite states. A composite state may contain nested states either concurrent or sequential which are called orthogonal substates and nonorthogonal substates respectively. Given a set of nonorthogonal substates in the context of an enclosing composite state called OR-state, the object is said to be in the composite state and in only one of those substates at a time [12]. In the case of orthogonal substates, the concept of region is introduced which specifies each state machine that execute in parallel in the context of the enclosing composite state called AND-state. Only one substate from each of the orthogonal regions is active as long as the object remains in the corresponding AND-state.

Initial state indicates the default starting place for the state diagram or substate while final state indicates that the execution of the state machine or the enclosing state has been completed. Another special state is the history state which allows an OR-state to remember the last substate that was active prior to the leaving from the OR-state.

Transitions are relationships between a pair of states indicating that an object in the first state will enter the second state when a specified event occurs under certain condition. Therefore, a transition t comprises three parts: source state denoted by $src(t)$ which is the state affected by the transition; target state denoted by $dst(t)$ which the object enters after the completion of the transition; label denoted by $EGA(t)$ which contains events, guards, and actions.

Semantics of transitions varies according to its source and target state. When leading out of a composite state, a fired transition leaves the active nested states before leaving the composite one. When targeting a composite state, a fired transition would lead the object to the initial state of each nested machine running in parallel after entering the composite state.

In addition to these regular transitions, there exist some special ones. **Completion transition** is a transition with no event trigger, the fire of which depends on the completion of the behavior within its source state. Transition **join** which sources multiple states allows the object to leave all the orthogonal regions of an AND-state at one time. Similarly, transition **fork** which targets multiple states enables passing directly to all the orthogonal regions of an AND-state. The initial state of the regions which have no target states of the **fork** will be activated.

With clear semantics of each element, the transformation mechanism which deals with different elements with different mapping rules can be determined.

3.3 FSM Model

Finite State Machines (FSM) are models each built with a set of states, as well as transitions going from one state to another, which are triggered either by inputs from outside or changes within the system itself. The execution would start from a state called start state and keep running until reaching a state called accept state. As its mathematic nature, we can establish a formal representation for FSM which is the target model during the transformation process for facilitating automation.

Definition1. A FSM (Finite State Machine) A is a quintuple (Q, L, δ, q_0, q) , where Q is a finite set of states of A , L is a finite set of transition labels of A , $\delta: Q \times L \rightarrow Q$ is the transition function relating two states by the transition going between them, $q_0 \in Q$ is the start state, $q \in Q$ is the accept state.

If transition $t \in \delta$ represented as (s, l, s') , then **source** (t) = s , **target** (t) = s' , **label** (t) = l .

3.4 Transformation from State Diagram to FSM Model

As can be seen from the definition of FSM model, states involved are all basic ones, indicating that the removal of hierarchy is needed during the transformation process. For the sake of being conformed to the semantics of original models, the hierarchical relations between states of the state diagram should be obtained as critical information for generating corresponding FSM model without hierarchy. We therefore take the translation of topological structures of state diagrams to mathematic models of Hierarchical Finite State Machines (HFSM) as a preliminary step towards model transformation due to the fact that HFSM provides a simple and precise manner to illustrate the topological structure of a state diagram.

Different from FSM, HFSM contains states with inner structures. We could take HFSM as parallel and/or hierarchical composition of FSMs with states of higher hierarchy representing FSMs of lower hierarchy. A definition of HFSM is given below according to this point of view.

Definition2. Given a finite set of FSMs $F = \{A_1, \dots, A_n\}$ with mutually distinct state spaces $Q(A_i)$,

- $\phi: \bigcup_{A \in F} Q(A) \rightarrow P(F)$ is a composition function on F iff
- $\exists_1 A \in F \wedge A \notin \bigcup \text{ran}(\phi)$, which indicates a unique root FSM denoted by ϕ_{root}
 - $\forall A \in \bigcup \text{ran}(\phi) \bullet \exists_1 s \in \bigcup_{A' \in F \setminus \{A\}} Q(A') \bullet A \in \phi(s)$
 - $\forall S \subseteq \bigcup_{A \in F} Q(A) \bullet \exists s \in S \bullet S \cap \bigcup_{A \in \phi(s)} Q(A) = \emptyset$.

Definition3. Hierarchical finite state machine (HFSM) is a pair (F, ϕ) where F is a set of FSMs with mutually distinct state spaces, ϕ is a composition function on F .

With the definition of HFSM, the topological structure of the original state diagram could be obtained in a formal representation, which is specified by the composition function ϕ . Construction of such structure starts from the top hierarchy, and then gradually comes to completion by detailing each composite state that belongs to the state diagram level by level. Establish $\phi(s) = A_i$ and $F = F \cup \{A_i\}$ if the composite state s is an OR-state with a sub-machine A_i enclosed, while $\phi(s) = \{A_1, A_2, \dots, A_n\}$ and $F = F \cup \{A_1\} \cup \{A_2\} \cup \dots \cup \{A_n\}$ if the composite state s is an AND-state with sub-machines A_1, A_2, \dots, A_n each located in the corresponding orthogonal region of s . The state pointed by initial state turns to be the start state of the corresponding FSM, while the state which points at final state becomes the accept state.

Once the representation for topological structure is present, we can get to know the hierarchical relation between states which can be specified by the following function. When given a HFSM (F, ϕ) :

$$\chi: \bigcup_{A \in F} Q(A) \rightarrow P(\bigcup_{A \in F} Q(A))$$

$$\chi(s) = \{s' \mid \exists A \in F \bullet A \in \phi(s) \wedge s' \in Q(A)\}$$

With hierarchical information represented in mathematic form, the transformation to the resulting FSM model starts from that of transitions of the original state diagram. But some preliminary conceptions have to be introduced first.

Definition4. A set $C \subseteq \bigcup_{A \in F} Q(A)$ is a **configuration** of a given HFSM (F, ϕ) iff

- $\exists_1 s \in Q(\phi_{\text{root}}) \bullet s \in C$
- $s \in C \wedge A \in \phi(s) \Rightarrow \exists_1 s' \in Q(A) \bullet s' \in C$
- $s \in C \wedge \exists s' \bullet s \in \chi(s') \Rightarrow s' \in C$

Definition5. Given a HFSM (F, ϕ) with C as the set of all its configurations and s as one of its states, function **config**: $\bigcup_{A \in F} Q(A) \rightarrow P(\bigcup_{A \in F} Q(A))$

$$\text{config}(s) = \{ci \mid ci \subseteq C \wedge s \in ci\}$$

Definition6. Given a HFSM (F, ϕ) , the **default configuration** of certain state sd is denoted as a function **deconfig**: $\bigcup_{A \in F} Q(A) \rightarrow P(\bigcup_{A \in F} Q(A))$

$$\text{deconfig}(sd) = X \Leftrightarrow \exists_1 X: \text{config}(sd) \bullet$$

$$\forall s \bullet (s \in X \wedge s \notin \chi^*(s) \Rightarrow \bigcap q_0(\phi(s)) \subseteq X)$$

Definition7. Given a state diagram with one of its transitions t , $Uexit$ is the uppermost one among the states of the set $exit = \{exit_i \mid \forall j: N \bullet src_j(t) \in \chi^*(exit_i) \wedge dst_j(t) \notin \chi^*(exit_i)\}$, $Uenter$ is the uppermost one among the states of the set $enter = \{enter_i \mid \forall j: N \bullet src_j(t) \notin \chi^*(enter_i) \wedge target_j(t) \in \chi^*(enter_i)\}$.

States of the resulting FSM model are configurations each represent a set of states of the original state diagram which are active at present. Therefore, transitions involved are running from one configuration to another, which leads to the fact that each transition of the state diagram may correspond to several transitions within target FSM model according to the number of configurations the source state of the original transition belongs to. Suppose $confTranSet$ is the transition set of the resulting FSM, the algorithm for obtaining the set is specified below:

```

for each transition  $t$ 
  if  $EGA(t) = \emptyset$ 
    TempSet =  $\bigcap q (\phi_i(src(t)))$ 
    for each  $q_i \in TempSet$ 
      configi = config( $q_i$ )
    ConfSet =  $\bigcap config_i$ 
    DefConf = deconfig( $dst(t)$ )
  if  $t$  is a join
    for each  $s_i \in src(t)$ 
      configi = config( $s_i$ )
    ConfSet =  $\bigcap config_i$ 
    DefConf = deconfig( $dst(t)$ )
  if  $t$  is a fork  $\wedge |dst(t)| > 1$ 
    ConfSet = config( $src(t)$ )
    defDst =  $\bigcup (deconfig(dst_i(t)) \cap \chi^*(dst_i(t)))$ 
    NdefDst =  $\bigcap (deconfig(dst_i(t)) \setminus \chi^*(dst_i(t)))$ 
    DefConf = defDst  $\cup$  NdefDst
  else
    ConfSet = config( $src(t)$ )
    DefConf = deconfig( $dst(t)$ )
  while (ConfSet is not empty)
    get a souconf  $\in ConfSet$ 
    tarconf =  $(souconf \setminus \chi^*(Uexit(t)) \cup (\chi^*(Uenter(t)) \cap DefConf))$ 
    source( $t'$ ) = souconf
    target( $t'$ ) = tarconf
    label( $t'$ ) =  $EGA(t)$ 
    confTranSet = confTranSet  $\cup \{t'\}$ 
    confSet = confSet  $\setminus \{souconf\}$ 

```

Then the state set can be generated by filling up with states related to each element of the transition set $confTranSet$. The initial and accept state of the resulting FSM model *InitState* and *AccState* can also be determined.

InitState = deconfig($q_0(\phi_{root})$)

AccState = config($q(\phi_{root})$)

This is the process during which state set of the original state diagram are mapping into that of the resulting FSM model. But there're some exceptions.

History states are not involved in the algorithm due to their different semantics with other common states; we handle them in a special way.

For each history state h referring to certain OR-state *Ors* with a state set HS composed of all its nonorthogonal substates, we build relations of the target states of transitions leading out of state *Ors* with each hs_i ($hs_i \in HS$). Relations, represented by transitions, should be established in pairs, indicating returning to the same state that was last active when leaving the enclosing OR-state. Suppose the target state of the transition leading out of *Ors* is *Htar*, and the label of the transition is denoted as l , for each hs_i ($hs_i \in HS$), a new transition labeled "*back* (hs_i)" is created with *Htar* and hs_i as its source and target state. With a transition set obtained by the method above, the problem is then turning into the transformation from each element of the set to its counterparts of the resulting FSM model. Meanwhile, existing transitions of the newly established FSM model which labeled l should be modified. Suppose t is a transition of the resulting FSM model labeled l , then $label'(t) = label(t) + s$ ($s \in source(t)$).

Till now, a FSM model carrying the same semantics with the original state diagram is constructed and completed.

4. An Example: Software Download

An example of state diagram is shown in Figure 1, which models a web application for *software download*. The life cycle of the web application starts from its main page (MP), then turns to download or search module according to the choice of users. When entering the download module, two entities will be triggered: a web page for illustrating the usage of the software about to download by a video clip, a dialog box for download operation.

According to the transformation mechanism we proposed, the topological structure of the state diagram should be captured first by constructing a HFSM model. The resulting HFSM model can be generated as follows:

$A_1: (\{S1, S2, S3, S4\}, \{l_1, l_2, l_3, l_4\}, \{(S1, l_1) \rightarrow S2, (S1, l_2) \rightarrow S3, (S2, l_3) \rightarrow S4, (S3, l_4) \rightarrow S4\}, S1, S4)$
 $A_2: (\{S5, S6\}, \{l_5\}, \{(S5, l_5) \rightarrow S6\}, S5, S6)$
 $A_3: (\{S7, S8, S9\}, \{l_6, l_7\}, \{(S7, l_6) \rightarrow S8, (S8, l_7) \rightarrow S9\}, S7, S9)$
 $A_4: (\{S10, S11, S12\}, \{l_8, l_9\}, \{(S10, l_8) \rightarrow S11, (S11, l_9) \rightarrow S12\}, S10, S12)$
 $\phi: \phi_{root} = \{A_1\}, \phi(S2) = \{A_2, A_3\}, \phi(S3) = \{A_4\}, \phi(S1) = \phi(S4) = \dots = \phi(S13) = \emptyset$
 $F = (\{A_1, \dots, A_4\}, \phi)$

Then, each transition of the exemplified state diagram could be transformed into several transitions of the resulting FSM model by the algorithm we proposed with the HFSM model above. The results are shown as follows where L_i indicates the transition of the state diagram which labeled l_i ; C_i indicates one of the configurations of the HFSM model.

- $L_1: C1 = \{ \text{root}, S1 \}, C2 = \{ \text{root}, S2, S5, S7 \},$
 $(C1, l_1) \rightarrow C2$
 $L_2: C3 = \{ \text{root}, S3, S10 \}, (C1, l_2) \rightarrow C3$
 $L_3: C4 = \{ \text{root}, S2, S6, S9 \}, C5 = \{ \text{root}, S4 \}, (C4,$
 $l_3) \rightarrow C5$
 $L_4: C6 = \{ \text{root}, S3, S11 \}, C7 = \{ \text{root}, S3, S12 \}, (C3,$
 $l_4) \rightarrow C5, (C6, l_4) \rightarrow C5, (C7, l_4) \rightarrow C5$
 $L_5: C8 = \{ \text{root}, S2, S5, S8 \}, C9 = \{ \text{root}, S2, S5, S9 \},$
 $C10 = \{ \text{root}, S2, S6, S7 \}, C11 = \{ \text{root}, S2,$
 $S6, S8 \}, C12 = \{ \text{root}, S2, S6, S9 \}, (C2, l_5) \rightarrow C10,$
 $(C8, l_5) \rightarrow C11,$
 $(C9, l_5) \rightarrow C12$
 $L_6: (C2, l_6) \rightarrow C8, (C10, l_6) \rightarrow C11$
 $L_7: (C8, l_7) \rightarrow C9, (C11, l_7) \rightarrow C12$
 $L_8: (C3, l_8) \rightarrow C6$
 $L_9: (C6, l_9) \rightarrow C7$
 $L_{10}: (C1, l_{10}) \rightarrow C8, (C1, l_{10}) \rightarrow C11$
 $L_{11}: C13 = \{ \text{root}, S13 \}, (C12, l_{11}) \rightarrow C13$
 $L_{12}: (C7, l_{12}) \rightarrow C2$
 $L_{13}: (C6, l_{13}) \rightarrow C13$

We can now generate the state set of the resulting FSM model, which is filled up with all the configurations mentioned above: $Q = \{ C1, \dots, C13 \}$. The initial state is $C1$ while the accept state is $C5$.

Finally, noticing there's a history state H within the state $S3$, we should add several new transitions to the transition set of the FSM model:

- $(C5, \text{"back (S10)"}) \rightarrow C3, (C5, \text{"back (S11)"}) \rightarrow C6,$
 $(C5, \text{"back (S12)"}) \rightarrow C7$

Meanwhile, transitions labeled l_4 should be modified into:

- $(C3, l_4 (S10)) \rightarrow C5, (C6, l_4 (S11)) \rightarrow C5, (C7, l_4$
 $(S12)) \rightarrow C5.$

5. Implementation of the Transformation Mechanism

As automatic testing is our final goal of model transformation, the implementation of such mechanism by computer itself is required. The method proposed in

this section can be applied to all the diagrams of UML model, only the transformation mechanism varies when dealing with different kinds. Since computers are unable to understand and analyze meanings conveyed by diagrams, texts carrying equivalent amount of information would help. Here, we choose XMI.

5.1 XMI

XML Metadata Interchange (XMI) is a standard that enables users to express objects using Extensible Markup Language (XML), the universal format for representing data on the WWW. As a bridge across the gap of objects and XML, it provides a standard mapping from objects defined by UML to XML, fulfilling object-oriented feature of both UML and programming languages. In addition, many mature tools supporting transformation from UML diagrams to corresponding XMI files are presented, such as argoUML. Therefore, XMI becomes the ideal textual representation of those UML diagrams.

5.2 Implementation Method

First of all, XMI files are needed which can be easily obtained as output of argoUML with inputs as UML diagrams. As shown in Figure 2, when receiving the resulting XMI, we extract semantics by recognizing different tags which indicate the location of information related to certain elements of the UML diagrams. Then, data structure based on the corresponding HFSM model could be constructed. With topological information provided by the data structure, mapping rule for transforming to FSM models works. Finally, resulting models are made to be hold in XML files with schema defined by ourselves.

A tool prototype has been developed to support our transformation mechanism and implementation method. It takes state diagrams carried by XMI files as inputs and resulting FSM model carried by XML files as output. Also, one can modify the chosen XMI file through an edition platform provided by the tool before transformation operation starts.

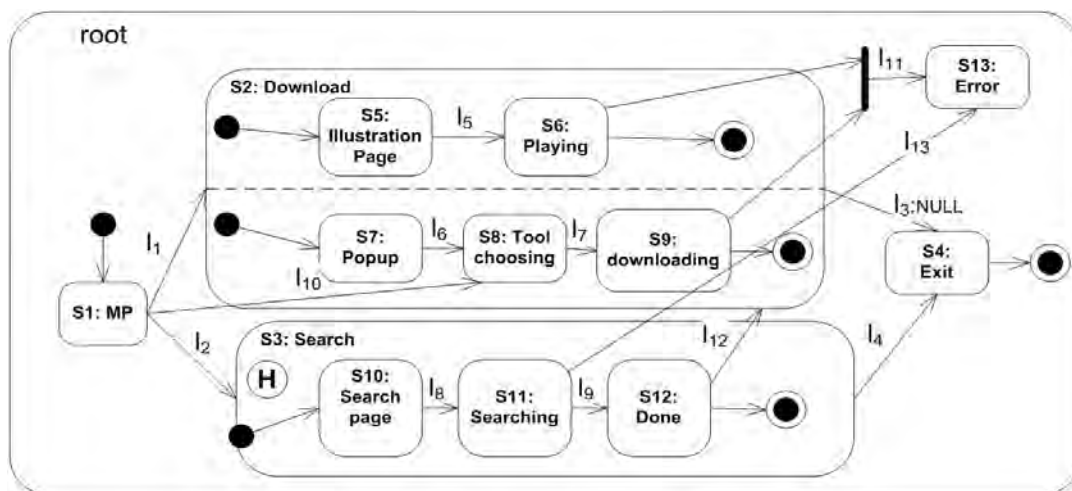


Figure 1. State diagram of a web application for software download

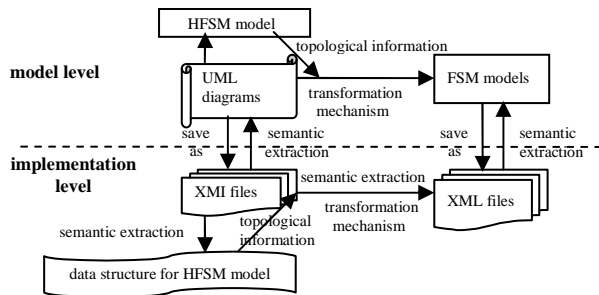


Figure 2. Implementation process

5.3 Simulation

For the purpose of verifying the correctness of our approach, we use the tool developed by ourselves to simulate the example presented in the previous section.

Figure 3 shows an interface of our tool for automatic testing for web applications. The characters in the main frame are the textual representation of the example state diagram.

After choosing transformation function of the tool, the model will then be transformed into FSM model written in XML language, as shown in Figure 4.

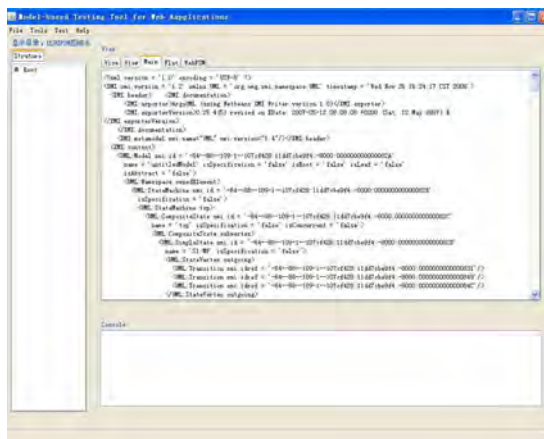


Figure 3. An interface of the tool for automatic testing for web applications

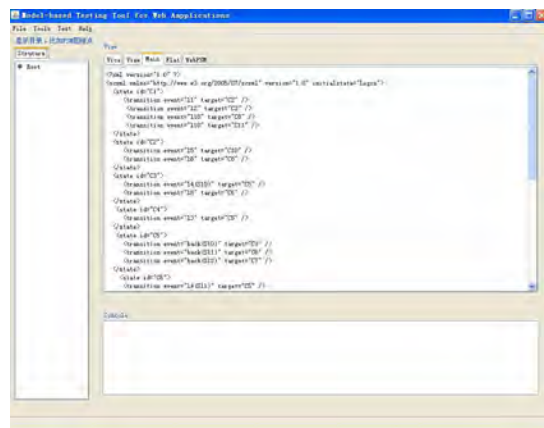


Figure 4. FSM model written in XML language

To illustrate the resulting FSM model more clearly, our tool implements the visualization of its textual representation, which can be seen in Figure 5.

6. Conclusions

This paper proposes a method for transformation from UML model to FSM model. It allows users to model a system with the language they used to without barriers towards automatic and efficient testing. As we focus on the translation of state diagrams, a specific transformation mechanism is proposed which enables generation of corresponding FSM models with same semantics.

Modelers create one state diagram for each object of the system and other UML diagrams for relations between them. Since our specific transformation mechanism serves for every single state diagram, synthesis of the FSM models each obtained from one of these state diagrams should be discussed. It depends on the information provided by other UML diagrams like class diagrams, sequence diagrams etc. Besides, these UML diagrams themselves need to be transformed into FSM with meta-model we defined so as to generate target model that covers information carried in all of the given UML models. They could either be transformed directly into FSM models, or to state diagrams as the first step, which would then come into FSM models by the mechanism we proposed. Experiments about comparison on efficiencies of both should be hold with complete transition mechanisms before the choice can be made.

Besides, details of elements contained in labels including event, guard and action, as well as the action attribute of states, are not considered in our research, their affections to the correctness of transformation is also a part of the future work.

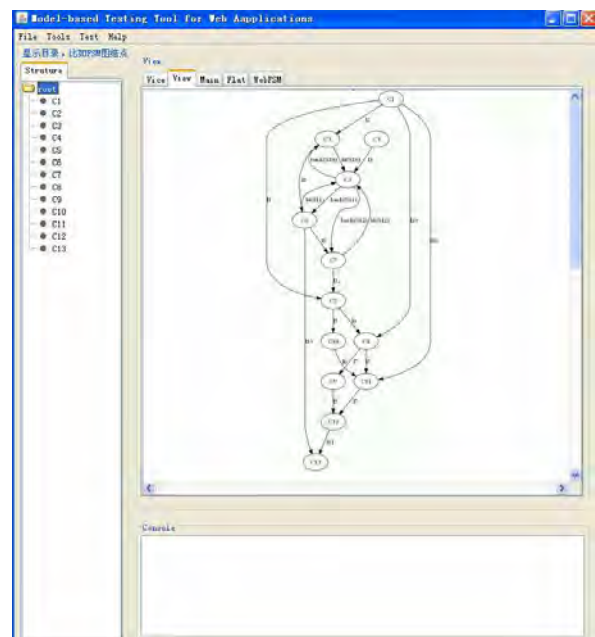


Figure 5. The visualization of the model's textual representation

7. Acknowledgement

This work has been supported by National High-Technology Research and Development Program of China under grant No. 2007AA01Z144, Natural Science Foundation of China under grant No. 60673115, National Grand Basic Research Program of China under grant No. 2007CB310800, Research Program of Shanghai Education Committee under grant No. 07ZZ06 and Shanghai Leading Academic Discipline Project, Project Number: J50103.

REFERENCES

- [1] S. R. Dalal, A. Jain, N. Karunanithi, and B. M. Horowitz, "Model-based testing in practice," Proceedings of the 21st International Conference on Software Engineering, Los Angeles, California, United States, pp. 285–294, May 1999.
- [2] H. Robinson, "Graph theory techniques in model-based testing," International Conference on Testing Computer Software, 1999.
- [3] W. E. McUmbler and B. H. C. Cheng, "A general framework for formalizing UML with formal languages," Proceeding of the 23rd international conference on Software engineering, Toronto, Canada, pp. 433–442, 2001.
- [4] J. Offutt, S. Y. Liu, A. Abdurazik, and P. Ammann, "Generating test data from state-based specifications," The Journal of Software Testing, Verification, and Reliability, pp. 25–53, 2003.
- [5] C. J. Mallery, "On the feasibility of using FSM approaches to test large web applications," May 2005.
- [6] The precise group, <http://www.cs.york.ac.uk/puml/>.
- [7] R. M. Borges and A. C. Mota, "Integrating UML and formal methods," Electronic Notes in Theoretical Computer Science, Elsevier Science Publishers, pp. 97–112, July 2007.
- [8] D. Latella, I. Majzik, and M. Massink, "Automatic verification of a behavioral subset of UML Statechart diagrams using the SPIN model-checker," Formal Aspects of Computing, pp. 637–664, 1999.
- [9] I. Traore, "An outline of PVS semantics for UML statecharts," Journal of Universal Computer Science, 2000.
- [10] M. Z. Lai and J. Y. You, "Formalize the time-extended UML state chart with timed automata," Computer Applications, pp. 4–6, August 2003.
- [11] E. Mikk, Y. Lakhnech, and M. Siegel, "Hierarchical automata as model for statecharts," Proceedings of the 3rd Asian Computing Science Conference on Advances in Computing Science, pp. 181–196, 1997.
- [12] G. Booch, J. Rumbaugh, and I. Jacobson, "The unified modeling language user guide," China Machine Press, Beijing, 2006.
- [13] R. Hower, "Web site test tools and site management tools," Software QA and Testing Resource Center, 2002.
- [14] Belinfante, L. Frantzen, and C. Schallhart, "Tools for Test Case Generation," Model-based Testing of Reactive Systems, Springer LNCS 3472, Springer-Verlag, pp. 391–438, 2005.
- [15] M. Utting, A. Pretschner, and B. Legeard, "A taxonomy of model-based testing," Technical Report 04/2006, Department of Computer Science, The University of Waikato (New Zealand), April 2006.
- [16] I. K. El-Far and J. A. Whittaker, "Model-based software testing," Encyclopedia of Software Engineering, Wiley-InterScience, Vol. 1, pp. 825–837, 2002.
- [17] M. Blackburn, R. Busser, and A. Nauman, "Why model-based test automation is different and what you should know to get started," in International Conference on Practical Software Quality and Testing, 2004.
- [18] B. Legeard, F. Peureux, and M. Utting, "Controlling test case explosion in test generation from B formal models," The Journal of Software Testing, Verification and Reliability, 14(2): pp. 81–103, 2004.
- [19] A. Pretschner, H. Lötzbeyer, and J. Philipps, "Model based testing in evolutionary software development," IEEE International Workshop on Rapid System Prototyping 2001, pp. 155–161, 2001.
- [20] T. Isakowitz, E. A. Stohr, and P. Balasubramanian, "RMM: A methodology for structured hypermedia design," Communication of the ACM, Vol. 38, No. 8, August 1995.
- [21] F. Coda, C. Ghezzi, G. Vigna, and F. Garzotto, "Towards a software engineering approach to web site development," Proceedings of 9th International Workshop on Software Specification and Design, Ise-Shima, Japan, April 16–18, 1998.
- [22] H. Gellersen and M. Gaedke, "Object-oriented web application development," IEEE Internet Computing, January–February 1999.
- [23] J. Conallen, "Modeling web application architectures with UML," Communications of the ACM, Vol. 42, No. 10, October 1999.
- [24] D. C. Kung, C. H. Liu, and P. Hsia, "An object-oriented web test model for testing web applications," First Asia-Pacific Conference on Quality Software, pp. 30–31, October 2000.
- [25] F. Ricca and P. Tonella, "Analysis and testing of web applications," Proceedings of the 23rd International Conference on Software Engineering, pp. 12–19, May 2001.

An Algorithm for Generation of Attack Signatures Based on Sequences Alignment

Nan Li, Chunhe Xia, Yi Yang, Hai-Quan Wang

State Key Laboratory of Virtual Reality Technology, Key Laboratory of Beijing Network Technology, School of Computer Science and Engineering, Beihang University, Beijing, China

Email: {linan, yangyi}@cse.buaa.edu.cn, {xch, whq}@buaa.edu.cn

Received November 11th, 2008; revised November 19th, 2008; accepted November 24th, 2008

ABSTRACT

This paper presents a new algorithm for generation of attack signatures based on sequence alignment. The algorithm is composed of two parts: a local alignment algorithm—GASBSLA (Generation of Attack Signatures Based on Sequence Local Alignment) and a multi-sequence alignment algorithm—TGMSA (Tri-stage Gradual Multi-Sequence Alignment). With the inspiration of sequence alignment used in Bioinformatics, GASBSLA replaces global alignment and constant weight penalty model by local alignment and affine penalty model to improve the generality of attack signatures. TGMSA presents a new pruning policy to make the algorithm more insensitive to noises in the generation of attack signatures. In this paper, GASBSLA and TGMSA are described in detail and validated by experiments.

Keywords: Attack Signatures Generation, Sequence Local Alignment, Affine Penalty, Intrusion Detection, Pruning Policy

1. Introduction

Network worms, viruses and malicious codes are still the top threat against the current Internet and enterprise security, and they cause a loss of hundreds of millions dollars every year [1]. Intrusion detection based on attack signatures is the most effective solution of this issue currently, but the continuous emergence of new types of attacks and polymorphic engines such as PHolyP [2] are great challenges to the existing intrusion detection technologies. To solve this problem, automatic generation of attack signatures has been concerned by more and more researchers and has become a new hotspot in intrusion detection since 2003 [3].

Algorithms for generation of attack signatures can be divided into two categories: one is based on string mode and the other is based on semantics. However, the latter relies on prior semantic analysis of a certain type of attacks, so it is incompetent for generating signatures of unknown attacks automatically. Currently the research on algorithms for generation of attack signatures is mainly based on string mode, including the following categories: algorithms based on the LCS (longest common substring), algorithms based on the Token (the strings appearing frequently in suspicious datum and containing more than one character) [4], algorithms based on sequence

alignment, algorithms based on finite automaton and algorithms based on protocol field and length [5].

The algorithms for generation of attack signatures based on Token is considered as the most effective and approbatory method currently. But in [3], the authors point out that signatures generated by this kind of algorithm are not precise and give out an algorithm based on sequence alignment. In this paper, we present a new algorithm for generation of attack signatures based on sequence alignment through analyzing the algorithms presented by [3] and referring to the idea of sequence alignment used in Bioinformatics. The algorithm is composed of two parts: GASBSLA algorithm and TGMSA algorithm. With the inspiration of sequence alignment used in Bioinformatics, GASBSLA replaces global alignment and constant weight penalty model by local alignment and affine penalty model to improve the generality of attack signatures. TGMSA presents a new pruning policy to make the algorithm more insensitive to noises in the generation of attack signatures.

The rest of the paper is organized as follows. Section 2 refers to related research, which describes the algorithms for generating attack signatures in [3] and analyzes its weakness. Section 3 presents the design of GASBSLA algorithm and TGMSA algorithm, and details their relative analysis. Section 4 presents the experiments on the effectiveness and the anti-noise ability of the algorithms. Section 5 concludes the paper and mentions of some future work.

This work was supported by three projects: the National 863 Project-Research on high level description of network survivability model and its validation simulation platform under Grant No.2007 AA01Z407, The Co-Funding Project of Beijing Municipal Education Commission under Grant No.JD100060630 and National Foundation Research Project.

2. Related Research

Sequence alignment is divided into pair-wise alignment and multi-sequence alignment, and most of multi-sequence alignment is based on pair-wise alignment. Firstly, this section introduces and analyzes a pair-wise sequence alignment algorithm CMENW (Contiguous- Matches Encouraging Needleman-Wunsch) and a multi-sequence alignment algorithm HMSA (Hierarchical Multi-Sequence Alignment) [3]. They are the most representative algorithms applied to the generation of attack signatures based on sequence alignment, and they are also the foundation of this paper. Then we introduce the most representative pair-wise local alignment algorithm—Smith-Waterman algorithm [6].

2.1 CMENW Algorithm

CMENW algorithm is a pair-wise alignment algorithm based on global alignment. It is improved on Needleman-Wunsch algorithm [7], which is the typical pair-wise alignment algorithm. The main difference between the two algorithms is: Needleman-Wunsch algorithm easily leads to fragments. In order to reduce the influence of fragments in the process of alignment, CMENW algorithm introduces contiguous-matches encouraging function $enc(x)$ (x is the number of contiguous bytes in the alignment), which is used to encourage contiguous bytes to be aligned together. The score function of CMENW algorithm is as follows:

$$S(x, y) = k_1 \times m + k_2 \times d + k_3 \times \delta + \sum enc(|s|) \quad (1)$$

m is the score of bytes matched, d is the score of bytes unmatched, δ is the score of empty penalty, k_1 is the number of bytes matched in the result of alignment, k_2 is the number of bytes unmatched, k_3 is the number of gaps, s is contiguous bytes.

Attack signatures generated by CMENW algorithm are not effective enough while facing to polymorphic attack because of the insufficient generality. It can be improved by using multi-sequence alignment, but the number of samples is difficult to meet the requirement in real world situation.

2.2 HMSA Algorithm

HMSA algorithm is a type of hierarchical multi-sequence alignment algorithm based on pair-wise alignment CMENW algorithm, which is suitable for attack signatures generation. This algorithm has three main features [3]: (1) hierarchical pair-wise alignment; (2) supporting wildcard characters; (3) with a pruning function.

HMSA algorithm possesses the function of pruning, which accelerates its convergence and enhances the noise resisting ability. However, the effectiveness of pruning function is based on two assumptions: (1) the alignment result of any two noise will be pruned because of trivial solution; (2) the alignment result of any two samples will not be pruned and get a precise attack signature. However,

in reality, it is possible that the alignment result of any two noises is not pruned, because input sequences of signatures generation algorithm are often processed by clustering algorithms. Thus the alignment results of noise that not pruned and the alignment results of sample will be easily prone to trivial solution and be pruned, and finally there is no result returned.

2.3 Smith-Waterman Algorithm

Smith-Waterman algorithm is a pair-wise local alignment algorithm put forward by Smith and Waterman in 1981, which is used to find and compare the similarity in local regions in an overall view. Even today it is still a common basic algorithm in bioinformatics. Given sequence x and y as inputs, Smith-Waterman algorithm outputs a local alignment result which is global optimal. The similarity value of it is maximal according to formula (2). And the meanings of the bytes in this formula are the same as those in the formula (1) in Section 2.1.

$$S(x, y) = k_1 \times m + k_2 \times d + k_3 \times \delta \quad (2)$$

Smith-Waterman algorithm is used to find the biggest similarity value and the best alignment based on the principle of dynamic programming, and its process includes two major steps:

1. Calculate the similarity values of two given sequences, and get a similarity matrix;
2. Get the best results of sequence alignment through dynamic programming and backtracking algorithm, according to the similarity matrix got in step 1.

Smith-Waterman algorithm improves Needleman-Wunsch algorithm. The main difference between them is: Smith-Waterman algorithm uses 0 to replace all the negatives in the similarity matrix; if the similarity values no longer increases when the length of alignment result increases, this algorithm will finish backtracking and output the result. According to the differences between the two algorithms, the idea of Smith-Waterman algorithm is helpful for CMENW algorithms to overcome the problem of insufficient generalization.

3. GASBSLA Algorithm and TGMSA Algorithm

Through the analysis of CMENW algorithm and HMSA algorithm, we present a new algorithm for generation of attack signatures based on sequence alignment. The algorithm is composed of two parts: a local alignment algorithm—GASBSLA (Generation of Attack Signatures Based on Sequence Local Alignment) and a multi-sequence alignment algorithm—TGMSA (Tri-stage Gradual Multi-Sequence Alignment).

3.1 GASBSLA Algorithm

In Bioinformatics, local alignment has more practical significance than global alignment because two sequences are often with very high similarity just in some local regions [8]. For example, two long DNA sequences often

have relation with each other only in seldom areas (password districts); proteins belonging to different families often have some regions in the same on the structure and function. The situation in generating of attack signatures is very similar with that of Bioinformatics, so GASBSLA algorithm replaces global alignment by local alignment to improve the generality and precision of attack signature under the conditions of a small sample. In addition, to further reduce the number of fragments, GASBSLA algorithm replaces constant weight penalty model by affine penalty model [9].

The differences between affine penalty model and constant weight penalty model are: the penalty for each gap is independent in constant weight penalty model. That is, in any case, the penalty for one gap is d , and the penalty for n gaps is nd ; but in affine penalty model, the penalty for n gaps which attached together is $q + (n-1) \times r$. Where q is the penalty for the first one of n gaps attached together, r is the penalty for the other gaps, and $r \ll q$. We can learn from descriptions above that in affine penalty model, the penalty for the first gap is more than the other ones which means the reduction of single gaps and fragments in the attack signatures.

The general idea of GASBSLA algorithm based on Dynamic Programming is: First, calculating the similarity values of two sequences and keeping them in a matrix (named similarity matrix or DP matrix); second, according to the dynamic programming backtracking algorithm, finding the optimal alignment sequence on the basis of the DP matrix. Both the time complexity and the space complexity of GASBSLA algorithm are $O(mn)$, where m and n are the lengths of the two sequences.

$\sigma(x, y)$ is the similarity value of the alignment of x and y , where x and y are any two characters.

Algorithm 1. GASBSLA algorithm

Input: sequence a and b

Output: the similarity value and optimal sequence alignment of a and b

Initialization:

a. $T(0,0) = 0$

b. **For each** $i = 1, 2, \dots, M$

$F(i,0) = 0, T(i,0) = 0$

c. **For each** $j = 1, 2, \dots, N$

$F(0,j) = 0, T(0,j) = 0$

Main iteration:

For each $i = 1, 2, \dots, M$

For each $j = 1, 2, \dots, N$

$$T(i,j) = \begin{cases} T(i-1,j-1), & \text{if } a_i = b_j \\ 0, & \text{if } a_i \neq b_j \end{cases}$$

$$F(i,j) = \max$$

$$\begin{cases} 0 & [\text{case } 1] \\ F(i-1,j-1) + \sigma(a_i, b_j) + \text{enc}(T(i,j)) & [\text{case } 2] \\ F(i-1,j) + \sigma(a_i, -) & [\text{case } 3] \\ F(i,j-1) + \sigma(-, b_j) & [\text{case } 4] \end{cases}$$

$$Ptr(i,j) = \begin{cases} STOP & [\text{case } 1] \\ DIAG & [\text{case } 2] \\ LEFT & [\text{case } 3] \\ UP & [\text{case } 4] \end{cases}$$

3.2 TGMSA Algorithm

TGMSA algorithm presents a new pruning policy to avoid the situation of no output caused by not being pruned in the alignment process of two noises. The general idea is modifying pruning policy in the n th ($n > 1$) layer alignment according to alignment similarity value. If the alignment similarity value is less than the threshold (that the alignment similarity value is out of confidence interval), the alignment result will not be pruned, but the two sequences will be laid aside then align respectively with the signature sequence result, which is the alignment result of other sequences. If the alignment result does not accord with pruning conditions, it will replace the original signature sequence, otherwise it will be deserted.

Algorithm 2. TGMSA algorithm

Input: sequence set S

Output: multi-sequence alignment result

Initialization:

$R \leftarrow S$

$W \leftarrow \{\}$

$T \leftarrow \{\}$

Iteration of the first stage:

while $|R| \geq 1$, do

if $|R| = 1$ (denote the sequence by s_i)

then $s_i \rightarrow W$

else

take out two sequences s_i and s_j orderly from R

align s_i with s_j using pair-wise alignment algorithm, the alignment result is denoted by Ali_{s_i, s_j} (including the similarity value and optimal sequence alignment)

pruning

if the number of fragments in $Ali_{s_i, s_j} \geq 3$ and there exists at least two fragments whose length ≥ 3

$Ali_{s_i, s_j} \rightarrow T$

Iteration of the second stage:

do

$V \leftarrow \{\}$

while $|T| \geq 1$, do

if $|T|=1$ (denote the sequence by s_i)
 then $s_i \rightarrow V$
 else
 take out two sequences s_i and s_j randomly from R
 align s_i with s_j using pair-wise alignment
 algorithm, the alignment result is denoted by Ali_{s_i, s_j}
 pruning
 if the similarity value of Ali_{s_i, s_j} falls in confidence
 interval (the calculation of similarity value confidence
 interval will be specified in Section 3.3.)
 $Ali_{s_i, s_j} \rightarrow V$
 else $Ali_{s_i, s_j} \rightarrow W$
 $T \leftarrow V$
 until $|V| \leq 1$
Iteration of the third stage:
 if $|V|=1$
 while $|W| \neq 0$
 take out single sequence from W orderly, then align
 it with the alignment result Ali in the second stage
 respectively to generate a new alignment result Ali' if
 the number of fragments in $Ali' \geq 3$ [10] and there exists
 at least two fragments whose length ≥ 3 [11,12]
 then $Ali = Ali'$
 else $Ali = \Phi$

3.3 The Selection of Alignment Similarity Confidence Interval

Central limit theorem holds that regardless of the statistics population on the subject obeying whatever distribution, the distribution of sample mean is close to a normal distribution, the mean of normal distribution equals that of population distribution, and the variance equals that of population distribution divided by the Sample size. Therefore, we can estimate the average signature alignment similarity based on a certain attack by the average of the similarity value samples. We use all the alignment similarity values calculated in the first stage as a sample to calculate the similarity value confidence interval which is the judgement condition of pruning in the second stage.

Assume (F_1, F_2, \dots, F_n) is a sample of the alignment similarity value population F , so the sample mean and sample standard variance are as follows:

$$\overline{F_{n+1}} = \frac{\sum_{i=1}^{n+1} F_i}{n+1} \quad (3)$$

$$\begin{aligned} S_{n+1} &= \sqrt{\frac{1}{n} \left(\sum_{i=1}^{n+1} (F_i - \overline{F})^2 \right)} \\ &= \sqrt{\frac{1}{n} \left[\sum_{i=1}^{n+1} F_i^2 - (n+1) \overline{F_{n+1}}^2 \right]} \end{aligned} \quad (4)$$

According to the small probability event theory of normal distribution: the most datum of normal population (99.7%) falls in the range of $\mu \pm 3\sigma$, and those cases out of the range are called small probability events. Statistics holds that small probability events occur almost impossibly, and they can be ignored. The confidence interval of alignment similarity value is as follows:

$$\left[\overline{F_{n+1}} - 3 \frac{S_{n+1}}{\sqrt{n+1}}, \overline{F_{n+1}} + 3 \frac{S_{n+1}}{\sqrt{n+1}} \right] \quad (5)$$

That is:

$$\left[\frac{\sum_{i=1}^{n+1} F_i}{n+1} - 3 \frac{\sqrt{\frac{1}{n} \left(\sum_{i=1}^{n+1} F_i^2 - (n+1) \overline{F_{n+1}}^2 \right)}}{\sqrt{n(n+1)}}, \frac{\sum_{i=1}^{n+1} F_i}{n+1} + 3 \frac{\sqrt{\frac{1}{n} \left(\sum_{i=1}^{n+1} F_i^2 - (n+1) \overline{F_{n+1}}^2 \right)}}{\sqrt{n(n+1)}} \right] \quad (6)$$

4. Experimental Results

In this section we verify the effectiveness and the noise resisting ability by practical results. In our experiments, CMENW algorithm and HMSA algorithm are implemented to verify pertinently the effectiveness of improvement gave out in GASBSLA algorithm and TGMSA algorithm.

4.1 Experiments Environment

Hardware environment: Dawning Server (Intel® Xeon® CPU, 4G internal storage);

Software environment: Linux Red Hat 9.0 Operating System (the version of kernel is 2.4.20-8).

4.2 Algorithm Validity Verification

For the purpose of comparison, we selected the same experimental method as [3]. We generate signatures for polymorphic versions of four real-world exploits: Apache-Knacker [13], CodeRed II [14], IISPrinter [15] and TSIG [16]. The Apache-Knacker exploit, the CodeRed II exploit and the IISPrinter exploit use the text-based HTTP protocol. The TSIG exploit uses the binary-based DNS protocol. We use polymorphic engine to generate 150 samples for each exploit attack include 50 samples used to generate signatures and 100 samples used to detect false negatives. In order to simulate an ideal polymorphic engine, we fill wildcard and code bytes for each exploit with values chosen uniformly at random. In addition, we select 10,000 data samples without attacks from the MIT Lincoln Laboratory intrusion detection system test set—DARPA99 (the third week data sets) [17] to detect False positives.

In our experiments, we set the matching score $m = 1$, the mismatching score $d = -0.2$, the penalty for the first gap $\delta = -1$, the penalty for the other gaps $\delta' = -0.05$. The Contiguous-Matches encouragement function is set as:

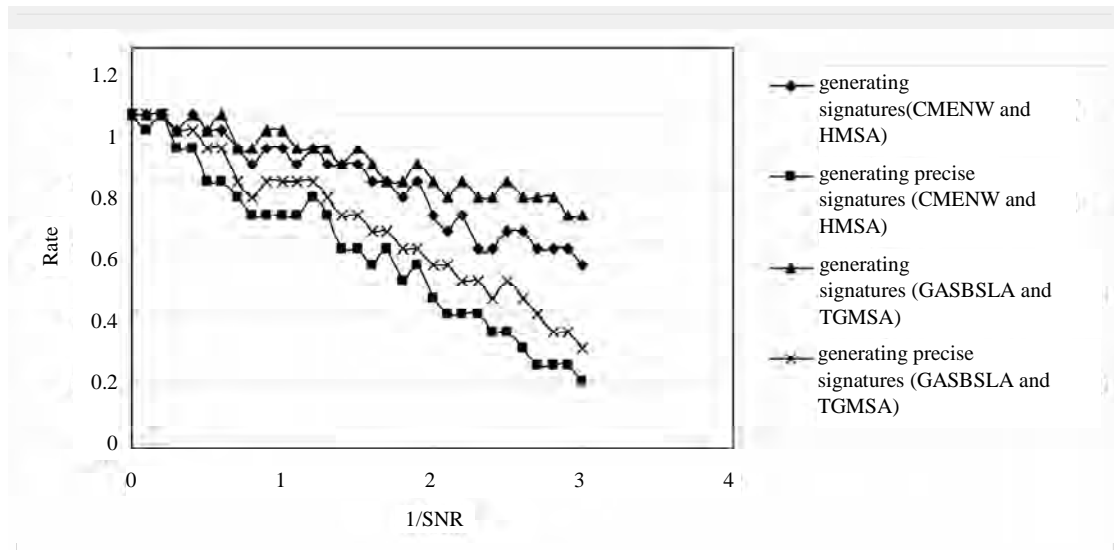


Figure 1. Rates of generating signatures and generating precise signatures for CodeRed II exploit attack in different SNR

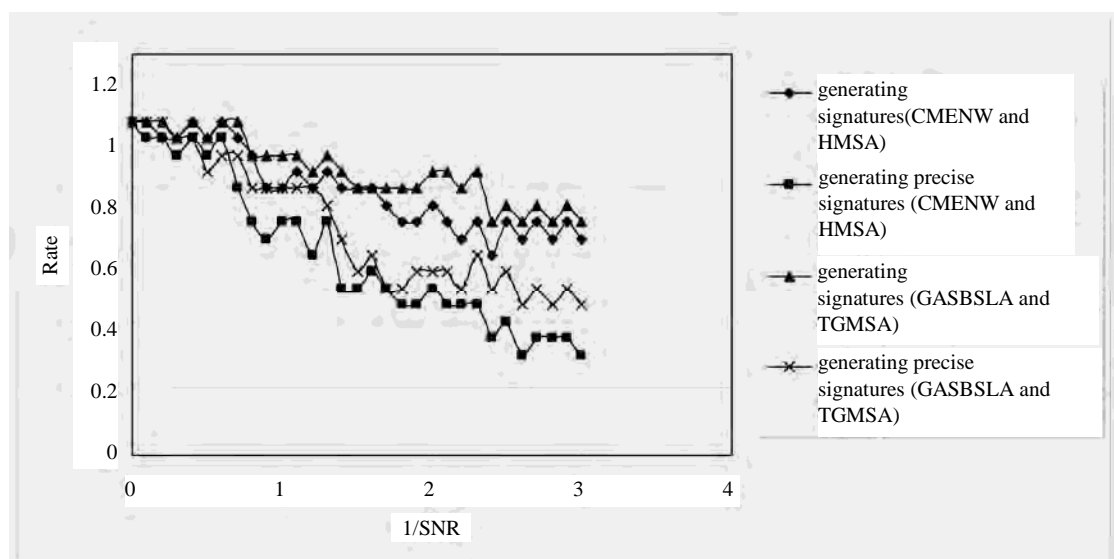


Figure 2. Rates of generating signatures and generating precise signatures for IISPrinter exploit attack in different SNR

5. Conclusions

In the paper, through analyzing the advantages and disadvantages of CMENW and HMSA algorithms we present a new attack signatures generation algorithm based on multi-sequence alignment with the idea of sequence alignment in bioinformatics. It contains two parts: a pair-wise local alignment algorithm-GASBSLA and a tri-stage gradual multi-Sequence alignment algorithm-TGMSA. GASBSLA algorithm uses the idea of local alignment and affine empty penalty model to improve the generality of attack signatures, so that it can detect polymorphic attack more effectively. TGMSA algorithm presents a new pruning policy to make the algorithm more insensitive to noises in the generation of attack signatures.

The experimental results indicate the advantages of the

algorithm as follows: the attack signatures result maintains a high degree of generality and a very good precision; it is more insensitive to noises in the condition that Signal-noise Ratio (SNR) is less than 1. The further study of our research mainly includes two parts: how to accelerate the convergence of TGMSA algorithm while maintaining the noise resisting ability; and how to improve the performance of the GASBSLA algorithm.

REFERENCES

- [1] Idc. IDC Enterprise Security Survey, 2005.
- [2] M. V. Gundy, D. Balzarotti, and G. V. Fieldschema, "Catch me, if you can: Evading network signatures with web-based polymorphic worms," Boston, MA: 2007.
- [3] Y. Tang, X. C. Lu, et al., "An automatic generation of attack signatures based on multi-sequence alignment [J],"

- Chinese Journal of Computers, 2006, 29 (9): 153321541.
- [4] J. Newsome, B. Karp, and D. Song, "Polygraph: Automatically generating signatures for polymorphic worms," in: Proceedings of the IEEE S & P 2005, Oakland, California, pp. 226–241, 2005.
 - [5] Z. Li, M. Sanghi, Y. Chen, et al., "Network-based and attack-resilient length signature generation for zero-day polymorphic worms[C]," 2007.
 - [6] T. Smith and M. Waterman, "Identification of common molecular subsequences," *Journal of Molecular Biology*, 147 (1): pp. 195–197, 1981.
 - [7] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology*, 48(3): pp. 443–453, 1970.
 - [8] P. K. Murphy, "Biological sequence comparison: An overview of techniques," Technical Report, University of Arizona, Department of Computer Science, 1994.
 - [9] S. Uliel, A. Fliess, A. Amir, and R. Unger., "A simple algorithm for detecting circular permutations in proteins," *Bioinformatics*, Vol. 15, No. 11: pp. 930–936, 1999.
 - [10] J. R. Crandall, S. F. Wu, and F. T. Chong, "Experiences using Minos as a tool for capturing and analyzing novel worms for unknown vulnerabilities," in: Proceedings of the GI SIG SIDAR Conference on Detection of Intrusions and Malware and Vulnerability Assessment, Vienna, pp. 32–50, 2005.
 - [11] J. R. Crandall, Su Zhen Dong, S. F. Wu, and F. T. Chong, "On deriving unknown vulnerabilities from Zero Day polymorphic and metamorphic worm exploits," in: Proceedings of the ACM CCS 2005, Alexandria, Virginia, pp. 235–248, 2005.
 - [12] J. Xu, P. Ning, C. Kil, Y. Zhai, and C. Bookholt, "Automatic diagnosis and response to memory corruption vulnerabilities," in: Proceedings of the ACM CCS 2005, Alexandria, Virginia, pp. 223–234, 2005.
 - [13] Symantec Security Response: CodeRed Worm. <http://www.sarc.com/avcenter/venc/data/codered.worm.html>.
 - [14] C. CAN-2003-0245. Apache apr-psprintf memory corruption vulnerability. <http://www.securityfocus.com/bi-d/7723/discussion/>.
 - [15] Viruslist.com: Net-Worm. Linux. Adm. <http://www.viruslist.com/en/viruses/encyclopedia?virusid=23854>.
 - [16] SANS Institute: Lion worm. <http://www.sans.o-rg/y2k/lion.htm>.
 - [17] R. P. Lippmann, D. J. Fried, I. Graf, et al., "Evaluating intrusion detection systems: The 1998 DARPA offline intrusion detection evaluation," in: Proceedings of the 2000 DARPA Information Survivability Conference and Exposition, Hilton Head, SC, 2: pp. 1012–1035, 2000.

Workflow Mining of More Perspectives of Workflow

Peng Liu¹, Bosheng Zhou²

¹School of Computer Science and Technology, Beijing University of Aeronautics and Astronautics, Beijing 100191, China, ²School of Computer Science and Technology, Beijing University of Aeronautics and Astronautics, Beijing 100191, China
Email: childbiggo@hotmail.com

Received November 27th, 2008; revised November 30th, 2008; accepted December 1st, 2008.

ABSTRACT

The goal of workflow mining is to obtain objective and valuable information from event logs. The research of workflow mining is of great significance for deploying new business process as well as analyzing and improving the already deployed ones. Many information systems log event data about executed tasks. Workflow mining is concerned with the derivation of a graphical process model out of this data. Currently, workflow mining research is narrowly focused on the rediscovery of control flow models. In this paper, we present workflow mining of more perspectives of workflow to broaden the scope of workflow mining. The mining model is described with GBMS's VPML and we present the entire model's workflow mining with the GBMS's VPML.

Keywords: Workflow Mining, GBMs, VPML

1. Introduction

Workflow technology continues to be subjected to on-going development in its traditional application areas of business process modeling and business process coordination, and now in emergent areas of component frameworks and inter-workflow, business-to-business interaction. Addressing this broad and rather ambitious reach, a large number of workflow products are commercially available, which see a large variety of languages and concepts based on different paradigms.

In 1993, the standardization organization of workflow technology—Workflow Management Coalition (WFMC) was built. The definition of workflow in WFMC is as follows: workflow is concerned with the automation of procedures where documents, information or tasks are passed between participants according to a defined set of rules to achieve, or contribute to an overall business goal [1]. So, all kinds of activities of enterprises are organized and corresponded by business processes modeling and defined business logic relation. Workflow model is the process model that can be executed in Computer and its performance can be analyzed with the executing result. In selection of workflow engine, it must review the analysis ability of the process model.

Workflow mining means the knowledge discovery of workflow system, which can be induced by the definition of data mining. The ultimate target of workflow mining is to mine the transactional logs of workflow system, and to discover the knowledge of workflow including workflow models mining, workflow performance mining and workflow models improving.

Most workflow mining research is aimed at the rediscovery of explicit control flow models, which are used to specify the behavior of a process. We believe that

this approach limits the scope and utility of workflow mining by neglecting the important notion that workflow is much more than control flow. In fact, the behavior of a process is but one of its perspectives. In [2], Wang Lei and Zhou Bosheng present four major perspectives (Behavior Model, Information Model, Resource Model, and Coordination Model) of a process model that have emerged from the disciplines that helped shape the workflow area.

2. The Research Basis

2.1 GBMS and VPML

VPML [3] (Visual Process Modeling Language) is a graphic language that supports a special process definition. Process is a set of transformations of input elements into products with specific properties, characterized by transformation parameters; VPML describes the process with visual process diagram and relevant text specification. The diagram shows the structure of the process and the specification shows the attribute of all items in the diagram. The process has a higher degree in visualization and formalization. The process model built in VPML can simulate. It was proved that VPML not only describes a whole complete process model, but also is a process modeling language with rich functions, visual diagram, easy learning, flexible application.

GBMS [4] (Government Business Modeling System) is a modeling system oriented E-government. GBMS not only supports the Government business's process handling, modeling, simulating and optimal restructure

that implements interdepartmental business integration and the sharing of information resources, but also organic integrates the government business modeling with the requirement extraction and analysis of the E-government Business Application System. GBMS completely describes the business model in the view aspect of process, organization, resource, information and collaboration and is process-centered, organic integrates five views and keeps each view's consistency and integrity. GBMS lays the foundation for design and implementation of the Business Application System in E-government area.

2.2 The Background and the System Architecture

The research background of this paper is that The Construction of Oriented E-government Software Component Library. First, using GBMS, we build process model, organization model, information model, collaboration model, resource model and behavior model. The process model is the core of these models, which clearly describes the government business and finds out the shared resources. Meanwhile by accumulating plenty of the government business models, it not only gradually establishes E-government business pattern base, designs and builds component library oriented E-government and sharing in departments, but also unifies the standard of the government business process and avoids the duplicate investment of E-government. On the base of the component library, it executes the models on jBPM's workflow engine and takes the rapid response to the new requirement in the E-government area, assembles the components of the E-government component library, and rapidly builds relevant the E-government business application system. It can build the system on demand and reduce production costs.

The system's architecture is given in Figure 1. The whole project has 5 parts:

(1) GBMS; (2) Component System; (3) jBPM Workflow Engine System; (4) Automatically Generating System; (5) Workflow mining System.

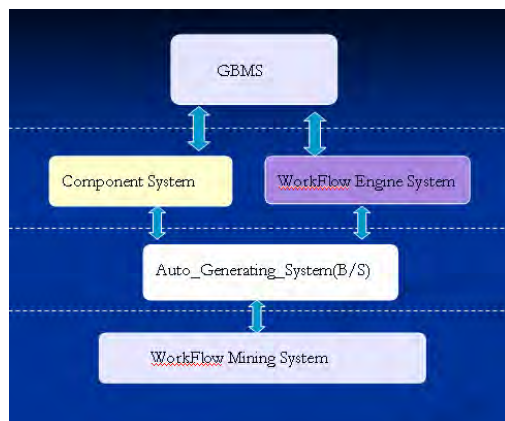


Figure 1. The system architecture

This paper extends the paper “*The Research on the Modeling Transformation From GBMS to jBPM*” [5], and presents how to mine the entire model from the E-government business system in order to validate the business model built in GBMS.

2.3 Semantics of GBMS and VPML

DEFINITION 1. GBMS as a 6-tuple

$PM = \{P, A, R, Control, Support, Input, Output\}$ where $P = \{Product_1, Product_2, \dots, Product_j\}$ is a set of products; $A = \{a_1, a_2, \dots, a_m\}$ is a set of Activities; $R = \{r_1, r_2, \dots, r_k\}$ is a set of resources; **Control** is a set of relations of controlling; **Support** is the relations of resources supporting activities, $Support \subseteq A \times R$; **Input** is the relations of activities and input products, $Input \subseteq A \times P$; **Output** the relations of activities and output products, $Output \subseteq A \times P$.

We need to define some assistant objects based on the above definition.

State (i): $i \in P \cup R$, represents the states of element i. It is enumeration type: $enum = \{able, disabled\}$ means the states can be either able or disabled.

Status (a): where $a \in A$, represents the status of the activity a. In GBMS Model, activity has 2 statuses, **START** and **END**.

Source(c): where $c \in Input \cup Output$, means the set of source objects related by c.

Target(c): where $c \in Input \cup Output$, is a set of target object related by c.

Prod_Source(a): where $a \in A$, represents the set of input products related with activity a.

Prod_Target(a): where $a \in A$, means the set of all output products related with activity a

Resource(a): where $a \in A$, the set of all resources related with activity a.

Role(a): where $a \in A$, represents the set of all roles related with activity a.

DEFINITION 2. The GBMS Data Model is used to describe the information perspective of workflow. GBMS Data Model is the 4-tuple $(P, A, Input, Output)$.

Input(p, a): is that the Activity a need Product p to run.

Output(p, a) is that Activity a is running to produce the Product p.

DEFINITION 3. The GBMS Organization Model is used to describe the organization perspective of workflow. GBMS Organization Model is the triple $(A, R, Support)$: A is the set of GBMS's Activities; R is the set of resources in GBMS; Support is the relations of resources supporting activities;

Support (a, r) is that the Resource r supports the Activity a to run.

DEFINITION 4. S is a subset of P, which represents the set of all source products.

$$S = \{S \subset P \mid \forall c \in \text{Input} \cup \text{Output}, \\ \text{Source}(c) \cap S = \Phi\}$$

DEFINITION 5. E is a subset of P , which represents the set of all the final products.

$$E = \{E \subset P \mid \forall c \in \text{Input} \cup \text{Output}, \\ \text{Target}(c) \cap E = \Phi\}$$

DEFINITION 6. $O(a)$ represents the constraints.

$$O(a) = \{a \in A, p \in \text{Prod_Source}(a), \\ r \in \text{Resource}(a), u \in \text{Role}(a) \mid \\ \text{Or } \text{State}(p) = \text{able} \ \& \ \& \ \text{State}(r) = \text{able} \\ \& \ \& \ \text{State}(u) = \text{able}\}$$

3. The Workflow Mining Algorithm

It is theoretically proved that VPML is equivalent Petri-Net [6]. So this paper uses the mining algorithm based on the Aalst's α -algorithm [7,8,9].

(1) Constructing dependence frequency table (D/F-table);

By given activities a and b , I) activity a and activity b 's appearance frequency $\#a$ and $\#b$; II) b is directly ahead of a : $\#b < \#a$; III) b directly succeeds a : $\#a > \#b$; IV) b is ahead of a : $\#b < \#a$; V) a is ahead of b : $\#a > \#b$; VI) the degree of dependence between a and b : $\#a \rightarrow \#b$;

(2) Mining activities relation table (R-table) by D/F-table;

Based on the D/F-table, mining the basic activities relations ($a >_w b$, $a \rightarrow_w b$, $a \#_w b$, $a //_w b$) [8];

(3) Reconstructing the workflow net by R-table and α -algorithm.

4. More Perspectives of Workflow Mining

In this section, we present a sample of more perspectives of Workflow Mining. The workflow event log is shown in Table 1. First, we do some reasonable assumptions because of the workflow mining algorithm. We assume that events are logged in temporal order, the event logs do not contain noise, and the event logs are theoretically complete.

4.1 Workflow Mining in the Behavior Model

The behavior model shows the behavior perspective of the workflow. This perspective is the basic and important of workflow. It is the performance of workflow system.

In the workflow event log table, it contains 3 cases. We use the algorithm in the 3rd section to mine the behavior model of GBMS.

(1) Definitude the basic relations between the activities using the D/F-table.

In the log, there are 3 cases, $\sigma_1 = \{A, B, C, D, E\}$, $\sigma_2 = \{A, B, D, C, E\}$, $\sigma_3 = \{A, F, G\}$.

Then we get the relations between activities:

$$a >_w b: A >_w B, A >_w F, B >_w C, B >_w D, C >_w D, C >_w E, D >_w E, \\ F >_w G;$$

$$a \rightarrow_w b: A \rightarrow_w B, B \rightarrow_w C, B \rightarrow_w D, A \rightarrow_w F, F \rightarrow_w G, C \rightarrow_w E, \\ D \rightarrow_w E;$$

$$a \#_w b: B \#_w F$$

$$a //_w b: C //_w D, D //_w C;$$

(2) Using the α -algorithm to mine the behavior model.

$$1) T_w = \{A, B, C, D, E, F, G\};$$

$$2) T_i = \{A\};$$

$$3) T_o = \{E, G\};$$

$$4) X_w = \{(A, B), (B, C), (B, D), (C, E), (D, E), (A, F), (F, G)\};$$

$$5) Y_w = X_w;$$

$$6) P_w = \{p(A, B), p(B, C), p(B, D), p(B, D), p(C, E), \\ p(D, E), p(A, F), p(F, G)\} \cup \{i_w, o_w\};$$

$$7) F_w = \{(A, p(A, B)), (p(A, B), B), (B, p(B, C)), \\ (p(B, C), C), (B, p(B, D)), (p(B, D), D), (C, \\ p(C, E)), (p(C, E), E), (D, p(D, E)), (p(D, E), E), \\ (A, p(A, F)), (p(A, F), F), (F, p(F, G)), (p(F, G), \\ G)\} \cup \{i_w, o_w\};$$

$$8) \alpha(W) = \{T_w, P_w, F_w\}.$$

(3) In GBMS, the behavior model is described in the Figure 2.

Table 1. Workflow event log

Case	Activity	Status	Need	Produced	Resource
1	A	START	DOC1		ROLE1
1	A	END		DOC2	
2	A	START	DOC1		ROLE1
2	A	END		DOC3	
3	A	START	DOC1		ROLE1
3	A	END		DOC2	
1	B	START	DOC2		ROLE2
1	B	END		DOC4, DOC5	
3	B	START	DOC2		ROLE2
3	B	END		DOC4, DOC5	
1	C	START	DOC4		ROLE4
3	D	START	DOC5		ROLE4
1	D	START	DOC5		ROLE5
3	C	START	DOC4		ROLE5
1	C	END		DOC6	
3	D	END		DOC7	
1	D	END		DOC7	
3	C	END		DOC6	
1	E	START	DOC6, DOC7		ROLE6
1	E	END		DOC9	
3	E	START	DOC6, DOC7		ROLE6
3	E	END		DOC9	
2	F	START	DOC3		ROLE3
2	F	END		DOC8	
2	G	START	DOC8		MACHINE1
2	G	END		DOC9	

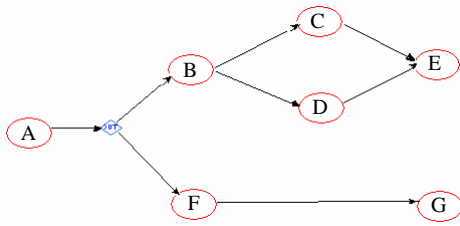


Figure 2. The GBMS behavior model

4.2 Workflow Mining in the Information Model

In workflow mining area, there have been no more researches in informational perspective. In this section, we introduce the workflow mining in the information model. The information model is used to describe the products consumed and produced in the business process and the relations between the products. In GBMS, the activity can be executed when it has products and roles that the states of those must be able to support. So the information model's mining is very important and can show the data flow of the process.

In the workflow event log Table 1, it contains 7 activities and 9 kinds of products. In activity's START status, it needs product to run; In END status, it can produce the new products. So in Table 1 of event log, we can get the formulas:

9) $\text{Prod_Source}(A)=\{\text{DOC1}\}; \text{Prod_Target}(A)=\{\text{DOC2}, \text{DOC3}\}$

10) $\text{Prod_Source}(B)=\{\text{DOC2}\}; \text{Prod_Target}(B)=\{\text{DOC4}, \text{DOC5}\}$

11) $\text{Prod_Source}(C)=\{\text{DOC4}\}; \text{Prod_Target}(C)=\{\text{DOC6}\}$

12) $\text{Prod_Source}(D)=\{\text{DOC5}\}; \text{Prod_Target}(D)=\{\text{DOC7}\}$

13) $\text{Prod_Source}(E)=\{\text{DOC6}, \text{DOC7}\}; \text{Prod_Target}(E)=\{\text{DOC9}\}$

14) $\text{Prod_Source}(F)=\{\text{DOC3}\}; \text{Prod_Target}(F)=\{\text{DOC8}\}$

15) $\text{Prod_Source}(G)=\{\text{DOC8}\}; \text{Prod_Target}(G)=\{\text{DOC9}\}$

Input={Input(A,DOC1), Input(B,DOC2), Input(C,DOC4), Input(D,DOC5), Input(E,DOC6), Input(E,DOC7), Input(F,DOC3), Input(G,DOC8)}

Output={Output(A,DOC2), Output(A,DOC3), Output(B,DOC4), Output(B,DOC5), Output(C,DOC6), Output(D,DOC7), Output(E,DOC9), Output(F,DOC8), Output(G,DOC9)}

Figure 3 illustrates the integration of the GBMS behavior model and data model.

4.3 Workflow Mining in the Organization Model

In GBMS, the organization model is used to define the government organization; it concludes all kinds of resources: person, machine, place, etc. In mining this aspect, a workflow analyst can discover if the participants of a business process are being used efficiently and effectively.

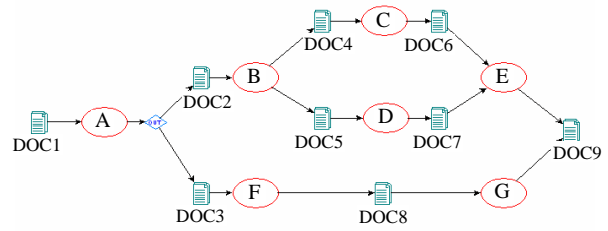


Figure 3. The rediscovered of GBMS behavior model and data model

In the workflow event log Table 1, it contains 7 activities, 6 roles and 1 machine. Activity needs the resource to support to run in the workflow system. From Table 1, we can get the formal representation of the rediscovered organization model:

1) Role (A)={ROLE1}

2) Role (B)={ROLE2}

3) Role (C)={ROLE4}

4) Role (D)={ROLE5}

5) Role (E)={ROLE6}

6) Role (F)={ROLE3}

7) Resource (G)={MACHINE1}

Support = {Support(A,ROLE1), Support(B,ROLE2), Support(C,ROLE4), Support(D,ROLE5),

Support(E,ROLE6), Support(F,ROLE3), Support(G,MACHINE1)}

The Figure 4 shows the integration of the GBMS behavior model, data model and organization model.

This sample is a business process of Beijing Xuanwu government shown in Figure 5. It is a process of business application. First the applicant fills in the table of application, and sends the table for the approval. If the application is the type of common service, then it is assigned to Windows' transaction, and distribute the app to different leaders to deal with, last it arrives the director to disposal. If the application is the type of administration permission, it is directly sent to the administrator to deal with and recorded into the database of Government. It was built by GBMS.

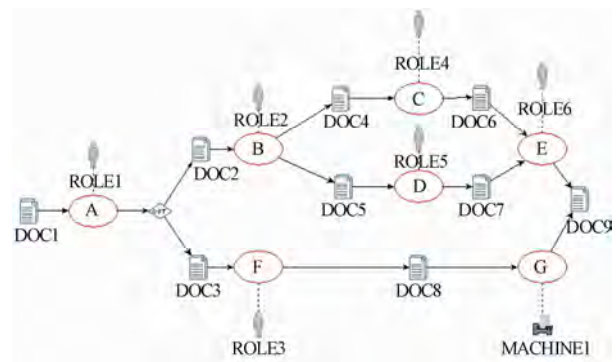


Figure 4. The rediscovered of GBMS behavior model, data model and organization model

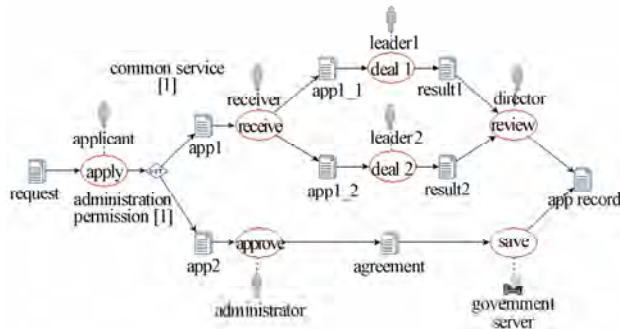


Figure 5. the application business process of GBMS

5. Conclusions and Future Work

Currently, most workflow mining algorithms have the goal of rediscovering a control flow model. We feel that this approach limits the scope and utility of workflow mining; it neglects the important point that workflow is much more than control flow. So this paper has presented the workflow mining in more perspective of workflow and given a full sample to show how to mine the entire GBMS model from the event log.

In future, we will study the incomplete log with noise and improve the mining algorithm. This is one research of the whole project. We also need to validate the mining model with the model design by the GBMS.

6. Acknowledgement

This project is funded by the Ministry of Science and Technology, P.R.C.

We should like to thank Beijing Cyber Technology for supporting the project.

REFERENCES

- [1] Workflow Management Coalition, "Workflow management coalition terminology and glossary," Technical Report, WfMCTC-1011, Brussels: Workflow Management Coalition, 1996.
- [2] L. Wang and B. S. Zhou, "The study of enterprise model," Computer Engineering and Application, 1002-8331-(2001)12-0005-05.
- [3] B. S. Zhou, H. X. and L. Zhang, "The Principle of Process Engineering and an Introduction to Process Engineering Environments," Journal of Software (supplement), pp. 519-534, August 1997.
- [4] B. S. Zhou and S. Y. Zhang, "Visual Process Modeling Language VPML," Journal of Software (supplement), pp. 535-545, August 1997.
- [5] P. Liu and B. S. Zhou, "The research on the modeling transformation from GBMS to jBPM," 2008 International Conference on Computer Science and Software Engineering.
- [6] A. H. Ren, "Research on the Concurrent Software Developing Method Based on Object Oriented Petri Nets," BHU, The school of computer science, pp. 116-128, 2001.
- [7] W. M. P. van der Aalst, T. Weijters, and L. Maruster, "Workflow mining: Discovering process models from event logs," IEEE Transactions on Knowledge and Data Engineering, 16(9), pp. 1128-1142, 2004.
- [8] A. J. M. M. Weijters and W. M. P. van der Aalst, "Process mining: Discovering workflow models from event-based data," in: B. Krose, M. de Rijke, G. Schreiber, M. van Someren (Eds.), Proceedings of the 13th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC 2001), pp. 283-290, 2001.
- [9] A. J. M. M. Weijters and W. M. P. van der Aalst, "Workflow mining: Discovering workflow models from event-based data," in: C. Dousson, F. Hooppner, R. Quiniou (Eds.), Proceedings of the ECAI Workshop on Knowledge Discovery and Spatial Data, pp. 78-84, 2002.

Complying with Coding Standards or Retaining Programming Style: A Quality Outlook at Source Code Level

Yanqing Wang, Bo Zheng, Hujie Huang

School of Software, Harbin Institute of Technology, Harbin 150001, P. R. China

Email: {yanqing, hjhuang}@hit.edu.cn

Received October 22nd, 2008; revised November 10th, 2008; accepted November 21st, 2008.

ABSTRACT

In order to make most software engineers and managers pay more attention to software quality at source level, two confusing terms—coding standard and programming style—were reviewed and compared. An evolutionary model of quality assurance at source code level was proposed, which implies that coding standard should be better accepted and more emphasized than programming style. Our current researches on evaluating the compliance with coding standards will likely make the strategy of quality assurance at source code level more operable.

Keywords: Coding Standards, Programming Style, Source Code, Quality Assurance (QA), Evaluation Index System

1. Introduction

How to raise the international competitive capability of software industry and how to improve software quality are new challenges for software companies and software engineers. The research on *coding standard* and *programming style* presents a quality notion of software engineering at source code level, while writing high quality code is a required skill to a software engineer. In this situation, how to make most software engineers and college students to pay more attention to quality at source level became an important subject in software industry and software engineering education field. In fact, complying with coding standards when programmers are writing computer programs is beyond the readability issue, it actually is an issue of quality assurance at source level. Since about 67% of workload in software lifecycle is at maintenance stage [1], coding standards can indirectly determine the quality of a whole project (see Figure 1).

Publications on this topic often focus on a set of rules on coding standards [2], taxonomy [3] and paradigm of programming style [4], and the approaches to teach coding standards [5]. These researchers upgraded the *coding standard* and *programming style* to high position at source code level. However, *coding standard* and *programming style* are not distinguished clearly in the above researches. Many software practitioners are always taking *coding standard* as the synonym of *programming style* and often use them alternatively. So it is very necessary to review, distinguish and compare these two confusing terms at first.

In this paper, the understanding of *coding standard* and

programming style was presented in Section 2 by analyzing their difference and relationship. In Section 3, an evolutionary model of quality assurance at source code level was constructed and put forward. The increasing importance of coding standards was emphasized in Section 4. In Section 5, our practices supporting the coding standards strategy were briefly introduced.

2. Understanding of Coding Standard and Programming Style

2.1 Explanation

(1) *Programming style*. It is also named *code style* or *coding style*. As we know, *programming style* is an intuitive and elusive concept that shows the style of writing code. It's highly individual and easily recognized, yet difficult to define or quantify. The goal of *programming style* is to make a program clear, easy to be understood, and thereby easy to work with [4], but the extreme personal *programming style* which is different from other team members' often degrades the readability of source code.

(2) *Coding standard*. Different from *programming style*, coding standards are a set of industry-recognized best practices that provide a variety of guidelines for developing software code. There is evidence to suggest that compliance with coding standards in software development can enhance team communication, reduce program errors and improve code quality [6]. The coding standards are not only assuring the readability, but also

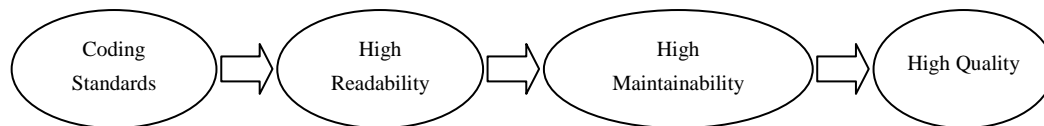


Figure 1. The relationship between coding standard and software quality

creating a helpful circumstance of teamwork. Working under a set of coding standards will make team members to comprehend colleague's programs much more easily and disabuse the injection of unworthy defects.

2.2 Difference

Programming style is not an equivalent of *coding standard*. Software engineers should understand their difference so that they can really control the quality of the programs they write.

(1) Different times. At early stage of software industry, software size was small, so one programmer can finish a whole project independently. The style of writing programs by individual programmer was called *programming style*. The programs at that time were more like crafts than engineering products, while the programmers were more like artists than engineers. With the rapid development of global IT industry, it was almost impossible for a single programmer to finish a whole software product. The production of software became the activity of teams, organizations or professional software companies. In software companies, various *programming styles* were combed and summarized to build up a set of rules—coding standards—to control the maintenance cost and software quality as well.

(2) Different measurability. The compliance with *coding standards* can be measured and evaluated but the *programming style* cannot be. Anyone who has different *programming style* from others can announce that he creates a new *programming style*. The *elegance* or *style* of a program is sometimes considered a nebulous attribute that is somehow unquantifiable; a programmer has an instinctive feel for a *good* or *bad* program [3]. It was believed that *programming style* is a multi-faceted concept that is not captured by a collection of rules or by a single style score [4]. It means that a program cannot be judged *good* or *bad* only by *programming style* or everyone can evaluate a program's style while he or she likes. On the contrast, the coding standards should be industry-recognized and almost software engineers recognized it [5]. Whether programmers complying with coding standards or not can be judged by human or by software programs.

(3) Different objective. The *programming style* focuses on more personal habits than readability. Some programmers retain a certain *programming style* because they are happy to do so. Different programmers may retain different *programming styles*. While coding standards emphasize readability and it prefers teamwork to individual. Writing program with coding standards improves the appearance of source code. Coding standards serves the teams and companies that care about the software quality at source code level.

2.3 Relationship

Coding standards are the evolution of *programming styles*. The *programming style* is an individual concept. As a *programming style* become popular and has been well accepted by many teams, companies, and even software industry, it will be upgraded to *coding standards*. With the development of software industry and outsourcing, the notion of *coding standards* has been taken on by more and more software enterprises. Over the past decades, with the international competition and the growing popularity of software outsourcing, international and industrial programming capabilities also have the unprecedented requirements for programmers. This was proved by the successful experience of software outsourcing to India [7]. As a result, coding standards usually need to be implemented through a formal process in industry [8] to assure the quality of source code.

Meanwhile, some scholars and engineers use these two terms alternatively. In [9], Kernighan and Plauger defined *programming style* as a set of rules or guidelines used when writing the source code for a computer program. They said, "It is often claimed that following a particular programming style will help programmers quickly read and understand source code conforming to the style as well as helping avoid introducing faults." From the above discussions in this paper, it is no doubt that the *programming style* in [9] is equivalent to *coding standard*.

3. Evolutionary Model of QA at Source Level

The software quality assurance contains five attributes: functionality, reliability, usability, performance, supportability [1]. Every attribute relates with source code. The quality of source code affects the quality of the software in large measure. To enhance the quality assurance at source level, an evolutionary model was constructed from the viewpoint of *coding standard* and *programming style* (see Figure 2).

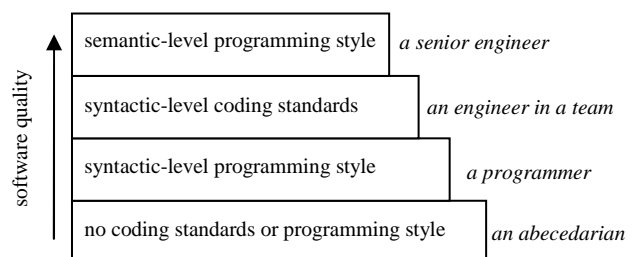


Figure 2. Evolutionary model of coding standard and programming style

Figure 2 embraces the process from lowest level *no coding standard or programming style* to the highest level *semantic-level programming style*. It also indicates the programmers who have the ability at corresponding level. The evolution is not only the time period concept but also a skill enhancement issue. Software quality will increase while its writer becomes a senior engineer from an abecedarian.

(1) Stage 1: no coding standards or programming style

At the very beginning of computer programming age, there was no *coding standards* or *programming style*. Programmers wrote programs as they wished. So did the students in colleges. The source code at this level was always confusing and had low readability. The software quality at this stage was hard to control.

(2) Stage 2: syntactic-level programming style

With the development of software engineering, the *programming style* was used to publicize the named good habits [9]. And the *syntactic-level programming style* had been used to assure the readability or efficiency of programs. The *programming style* helped the beginners become programmers. But as it stands, the *programming style* is too individual to be widely used.

(3) Stage 3: syntactic-level coding standards

With the industrialization of software development, piles of *programming styles* had to be summarized and upgraded to assure the team working more successful. Then the *syntactic-level coding standards* were implemented in some software companies such as IBM and Sun. The *syntactic-level coding standards* are the *coding standards* we defined above.

(4) Stage 4: semantic-level programming style

Moreover, while complying with coding standards, some experienced software engineers might write their programs in their own ways, e.g. inputting from or outputting to files, handling the connection with database and controlling the virtual memory etc., to make programs more efficient, reliable and portable. This level of *programming style* is defined as *semantic-level programming style*, which is based on syntactic-level coding standards and beyond them.

4. Role of Coding Standards in Software Engineering

4.1 Coding Standards Help Assure the Quality of Software

Coding standards can assure the software quality at source code level, and will release testers' workload. A defect in source code level would cost about 4 times money to remove when it remains at testing stage [1]. The coding standards can help the white-box tester read the program easily and save the testing time. The less time and more efficient, the higher quality the software would be. When more engineers concern source code quality and comply with coding standards, the quality of the softwares they produce would be assured.

4.2 Advices on Quality Assurance at Source Code Level

When the student starts to learn a new programming language or software engineers does their daily coding work, there are some advices based on the evolutionary model shown as above.

(1) If you are a student or novice software engineer, comply with *coding standards* and not retain *programming style* at syntactic level. Complying with *coding standards* is a basic skill of software engineer, especially working in a big company or developing an outsourcing project.

(2) If you are a teacher of a preliminary programming language, ask students to obey *coding standards*, educate tomorrow's qualified engineers at today's colleges.

(3) At the basis of complying the *coding standards*, develop your own *semantic-level programming style* when you are adequately experienced and you like. That is the highest realm of a programmer.

5. Conclusions and Future Work

Coding standard and *programming style* is a pair of terms at source code level. With the globalization of software cooperation and popularity of software outsourcing, the quality assurance at source level remains a pressing concern. Quality assurance at source level is much more economical and scientific than testing even though it is hard to operate. In this paper, *coding standard* and *programming style* were compared and an evolutionary model base on their relationship was proposed. *Coding standards* were more recommended than *programming style*. If we would like to retain our *programming styles*, the high level *programming style* beyond *coding standards* was strongly recommended.

To better support our ideas about complying with *coding standards*, we made some preliminary researches such as (1) we proposed an AHP-based evaluation index system on complying with coding standards [10,11]; (2) in order to make sense how many students were ready to write quality programs complying with coding standards, we did a case study and found that the results were not so satisfactory [12] because of the lack of consistent training and timely feedback; (3) a web-based evaluating platform was constructed, with which students can upload their programs anytime and get benchmarking results and detailed shortcomings of their programs on coding standards; (4) based on our previous work, a teaching model of coding standards based on evaluation index system and evaluating platform was proposed [13]. Even though we have made some progress, there are still lots of works to do in this subject. For example, a questionnaire designed for software industry should be delivered to make the evaluation index system more practical. Also, the teaching model should be refined with the accumulation of our experiences on programming languages teaching. Programming is a kind of art [14] so that it takes time for every engineer to reach the apogee of software quality.

REFERENCES

- [1] R. S. Pressman, Pressman, "Software engineering: A practitioner's approach," (6e), McGraw-Hill, 2005.
- [2] S. Herb and A. Andrei, "C++ coding standards 101 rules, guidelines, and best practices," Pearson Education Asia Ltd, 2006.
- [3] P. W. Oman and C. P. Cook, "A taxonomy for programming style," pp. 244–250, 1990, <http://doi.acm.org/10.1145/100348.100385>
- [4] P. W. Oman and C. P. Cook, "A paradigm for programming style research," SINGPLAN notices, Vol. 23, No. 12, 1990.
- [5] P. E. Berry and B. A. E. Meekings, "A style analysis of C programs," Communications of ACM, No. 28, pp. 80, January 1985.
- [6] X. S. Li and C. Prasad, "Effectively teaching coding standards in programming," in Proceedings of SIGITE'05, Newark, New Jersey, USA, October 20–22, 2005.
- [7] W. Kobitzsch, D. Rombach, and R. L. Feldmann, "Outsourcing in India," Software, IEEE, Vol. 18, No. 2, pp. 78–86, March–April 2001.
- [8] TIOBE Software BV, "TIOBE Coding Standard Methodology," 2003, Accessed March 4, 2005. <http://www.tiobe.com/standards/tekst.htm>
- [9] B. W. Kernighan and P. J. Plauger, "The Elements of Programming Style (2e)," McGraw Hill, New York, ISBN 0-07-034207-5, 1978.
- [10] Y. Q. Wang, J. Z. Wang, et al., "Quantitative research on how much students comply with coding standard in their programming practices," in the Proceedings of the 3rd China Europe International Symposium on Software Industry Oriented Education (CEIS-SIOE'2007), Dublin, Ireland, pp. 116–119, February 6–7, 2007.
- [11] Y. Q. Wang, J. Z. Wang, et al., "A framework for quantitative evaluation of coding standards in programming language teaching," Journal of Hefei University of Technology (Social Science), Vol. 3, pp. 67–71, 2008. (In Chinese)
- [12] Y. Q. Wang, H. D. Su, et al., "How many students are ready to write quality programs complying with coding standards: A case study," in the Proceedings of the 4th China Europe International Symposium on Software Industry Oriented Education (Guangzhou, China, January 10–11, 2008). CEISIE'2008. Zhongshan Daxue Xuebao/Acta Scientiarum Natralium Universitatis Sunyatseni, Vol. 46, No. SUPPL, pp. 93–96, December 2007.
- [13] Y. Q. Wang, L. Lei, et al., "Teaching model of coding standards based on evaluation index system and evaluating platform," in the Proceedings of 2008 International Conference on Computer Science and Software Engineering (Wuhan, China, December 12–14) CSSE'2008. IEEE Computer Society. (to be published)
- [14] D. E. Knuth, "The art of computer programming," Addison Wesley, 1999.