SEAMAIL

Newsletter from Software Engineers Association

Vol. 13, Number 9-10 January, 2003

目　次

編集部から

☆

この号は，昨年10月の ISFST-2002 で発表された論文のサマリーを特集しました．

☆☆

今度の ISFST では，これまでのように事前に Proceedings を印刷製本するのではなく，会議終了後に，実際に発表された論文だけを収録した Proceedings を，**紙ではなく CD で作る** という新機軸を試みました．

☆☆☆

この SEAMAIL に集めた各論文２ページずつのサマリーは，もともと会場で配付する計画だったのですが， Local Arrangement の手違いで，それが行われなかったので，あらためて SEAMAIL として，全会員に配付することにしました．

☆☆☆☆

なお，すべての Full Paper および Keynote Presentations のスライドのコピーを収録した CD (ここに印刷したサマリーも含まれています)を，付録として添付しましたので，それぞれの発表論文の詳細についてはそちらを御覧ください．

☆☆☆☆☆

面倒な編集作業に時間を割いてくださったプログラム委員長の塩谷和範さんに深く感謝します．

☆☆☆☆☆☆

Proceedings

The 7th International Symposium on Future Software Technology

---

ISFST-2002

October 20-21, 2002 Tutorials at Xi'an, China

October 23-25, 2002 Symposium at Wuhan, Chiina

Summaries of presented paper

Sponsored by

Software Engineers Association, Japan

(http://www.sea.jp/)

United Nations University / International Institute for Software Technology, Macau

Northwest University, Xi'an

Huazhong University of Science and Technology, Wuhan

ISFST-2002 Proceedings: ISBN 4-916227-14-X (CDROM)

Additional copies may be ordered from:

Software Engineers Association,

Marusho building 3-12 Yotsuya, Shinjuku, Tokyo, Japan,

Postal-Number: 106-0004

http://www.sea.jp

sea@sea.or.jp

Fax: +81-3-3356-1072

The International Symposium on Future Software Technology (ISFST) has been planed, funded and managed by the Software Engineers Association since 1996.

Editorial production by Kazunori SHIOYA
Program Co-chair of ISFST-2002
Compiled, Edited and Made in Japan.

Foreword

This is the 7-th symposium dedicated to the future software technology, which has been held in China since 1996. It has been organized by SEA (Software Engineers Association of Japan), UNU/IIST (International Institute of Software Technology of United Nations University), and the local host academic organizations in China.

This year, Northwest University (NWU) in Xi'an and Huazhong University of Science and Technology (HUST) in Wuhan are our host.

The purpose of the Symposium is to review the recent advances in software technology, to exchange various technical and business information and to promote the sound industry-academia cooperation in future. It will provide a unique opportunity for practitioners, researchers and educators in software community to gather latest information and discuss the trends in the field.

Pre-Symposium Tutorials in Xi'an: Oct.20 - 21

This year, prior to the symposium at Wuhan, a series of pre-symposium tutorials on several hot topics will be held in Xian. The city is a famous ancient capital and now a center of high technology in western region of China.

The Symposium at Wuhan: Oct.23 - 26

At the PC meeting held in Lushan in August, 49 papers were selected from about 150 submissions for technical presentation.

The symposium features two keynote speakers: Professor Alan Davis (University of Colorad, USA) and Professor Keijiro Araki (Kyushu University, Japan).

The unique feature of the ISFST symposium is that the program includes informal but intensive working-group style discussion sessions besides regular paper presentations. We have provided poster sessions for this purpose this year.

Welcome to ISFST 2002.

Kazunori SHIOYA

Soo-Yong Park

Yanxiang He

Program Co-Chairs

Symposium Organization

Honorary Co-Chairs:

Huowang Chen (National Univ. of Defense Technology, China)
Toshiharu Yamasaki (Former Chairman of SEA, Japan)

Organizing Co-Chairs

Hiroyuki Fukase (Chairman of SEA, Tokyo)
Kegang Hao (Northwest Univ., Xian)
Guohui Li (HUST, Wuhan)

General Co-Chairs

Akira Kumagai (PFU, Tokyo)
Yunsheng Liu (HUST, Wuhan)

Program Co-Chairs

Kazunori Shioya (SRA-KTL, Tokyo)
Soo-Yong Park (Sogang Univ., Seoul)
Yanxiang He (Wuhan Univ., Wuhan)

Program Committee

Keijiro Araki (Kyushu Univ., Fukuoka)
Doo-Hwan Bae (KAIST, Korea)
Doo-Kwon Baik (Korea Univ., Seoul)
Daoxu Chen (Nanjing Univ., China)
Zhiguo Chen (Henan Univ., China)
Guozhong Dai (IS-CAS, China)

Shihai Dong (Peking Univ., China)

Ming Fan (ZhengZhou Univ., China)

Chris George (UNU/IIST, Macau)

Yuqing Gu (ASTI Beijing, Beijin)

Jiang Guo (California State Univ., USA)

Hyuk-Soo Han (SangMyung Univ., Korea)

Jinpeng Huai (BUAA, China)

Hoh Peter In (Texas A&M Univ., USA)

Masao Ito (Nil Software, Tokyo)

Dehua Ju (ASTI Shanghai, Shanghai)

Kyo C. Kang (POSTECH, Korea)

Kiwamu Kase (RIKEN, Japan)

Gargi Keeni (TCS, India)

Chang-Hwa Kim (KangNung National Univ., Korea)

Jung-Ah Kim (KwanDong Univ., Korea)

Yue-Sun Kuo (Academia Sinica, Taipei)

Kei Kurakawa (NAIST, Nara)

Danning Li (Guizhou Academy of Sciences, China)

Yoshitaka Matsumura (SRA-KTL, Tokyo)

Deependra Moitra (Lucent Technologies, India)

Hideo Nakano (Osaka City Univ., Osaka)

Takamasa Nara (Hitachi S&S, Osaka)

Toshitsugu Nomura (JIPS, Tokyo)

Young-Bae Oh (SuWon Women's College, Korea)

Soo-Hyun Park (KookMin Univ., Korea)

Wanchai Rivepiboon (Chulalongkorn Univ., Thailand)

Beijun Shen (ECUST, China)

Tetsuo Tamai (Univ. of Tokyo, Tokyo)

Jiro Tanaka (Univ. of Tsukuba, Japan)

Hae-Sool Yang (HoSeo Univ., Korea)

Chenqing Ye (Zhejiang Univ., China)

Jinyuan You (Shanghai Jiao Tong Univ., China)

Mingyi Zhang (Guizhou Academy of Sciences, China)

Songmao Zhang (IM-CAS, China)

Zhizhuo Zhao (Xiamen Univ., China)

Xichang Zhong (SEC-CAS, China)
Youliang Zhong (Hypersoft, Australia)
Zhiying Zhou (Tsinghua Univ., China)

Xi'an Tutorial staff:

Yunwei Dong (Xian Synchrobit, Xian)
Gong Xiaoqing (Northwest University, Xian)

Secretary General

Kouichi Kishida (SEA/SRA-KTL, Tokyo)
Rongrong Li (HUST, Wuhan)

Referee List

Akira Kumagai *
Beijun Shen
Chris George
Deependra Moitra
DooKwon Baik
Gargi Keeni
Hideo Nakano *
Hiroyuki Fukase *
JeongAh Kim
Jiang Guo
Jiro Tanaka
Ju Dehua *
Kazunori Shioya *
Kei Kurakawa *
Keijiro Araki
Kouichi Kishida *
KyoChul Kang
Li GuoHui *
Liu YunSheng *
Masao Ito *
Shihai Dong
SooHyun Park
SooYong Park *
Tetsuo Tamai *
Toshitsugu Nomura
Xiaoqing Gong *
Yangxiang He *
Yoshitaka Matumura
YouLiang Zhong
YoungBae Oh
Yuesun Kuo
YunWei Don

** Attended PC meeting at Lushan in August 8-10.*

This page is intentionally made blank

ISFST-2002

Tutorial Program at Xi'an

Management Track

Day 1: Oct. 20 (Sunday)

Understanding and Using SW-CMM for Process Improvement (full day)

Speaker: Sandip Bose (Tata Consulting Serves, India)

Day 2: Oct. 21 (Monday)

Best Practices in System Testing (half day)

Speaker: K.Mahesh Kumar and Gargi Keeni (Tata Consulting Serves, India)

SPI Activities in Japanese Software Industry

Speaker: Kazunori Shioya and Kouichi Kishida (SRA-KTL, Japan)

Technology Track

Day 1: Oct. 20 (Sunday)

Internet Security (half day)

Speaker: Hideo Nakano (Osaka City University, Japan)

A Feature-Oriented Method for Product Line Software Engineering (half day)

Speaker: Kyochul Kang and Jaejoon Lee (POSTECH, Korea)

Day 2: Oct. 21 (Monday)

Developer-Centered Software Development Tools

Speaker: Yunwen Ye (University of Colorado, USA)

What is UML? - From the Viewpoint of Software Engineering and Computer Science

Speaker: Akira Kumagai (PFU, Japan)

Report on ISFST 2002 Xi'an Tutorial

ISFST 2002 Xi'an Tutorial was held in Xi'an Software Park from October 20 to October 21. There were about 100 people participated in the Tutorial. Among these participants, about 20 were from outside China, including Japan, Korea, India, U.S.A, and Mexico. The others were from the IT companies, universities and institutes in Xi'an.

The opening ceremony of the Tutorial commenced in Synchro Building at 9 o'clock on October 20. Mr. Zhu Kexiao, vice president of Northwest University, gave the Tutorial an opening address. Mr. Kouichi Kishida, Ms. Li Xuemei, deputy major of Xi'an, Mr. Xu Kewei, director of Science and Technology Bureau of Xi'an, addressed the opening ceremony. In addition, Prof. Hao Kegang, board chairman of SynchroBit Company, and Prof. Zhou Mingquan, dean of Computer Science Department of Northwest University were invited to attend the opening ceremony.

After half hour's opening ceremony, the tutorial was performed in two parallel tracks, management track and technology track. The topics and speakers are as followings:

1. Opening Ceremony
2. Management Track
   (1) Understanding and Using SW-CMM for Process Improvement
   Speaker: Gargi Keeni from TCS, India
   (2) Best Practices in System Testing
   Speaker: K.Mahesh Kumar from TCS, India
   (3) SPI Activities in Japanese Software Industry
   Speakers: Kouichi Kishida and Kazunori Shioya from SRA-KTL, Japan
3. Technology Track
   (1) Internet Security
   Speaker: Mr. Hideo Nakano from Osaka City University, Japan
   (2) A Feature-Oriented Method for Product Line Software Engineering
   Speaker: Prof. Kyochul Kang and Mr. Jaejoon Lee from POSTECH, Korea
   (3) Developer-Centered Software Development Tools
   Speaker: Yunwen Ye from University of Colorado, USA
   (4) UML (What is UML? -- From the viewpoint of Software Engineering and Computer Science)
   Speaker: Akira Kumagai from PFU, Japan

The Tutorial provided an opportunity for the participants to gather latest information and discuss the advances and the trends in IT field. It achieved great success, especially in exchange of various technical and business information.

Gong Xiaoqing

Northwest University, Xian

Note: *Tutorial materials are not included in this after proceedings in the CDROM.*

ISFST-2002

Final Technical Program *(presented papers only)*

Day 1: Oct. 23 (Wednesday)

08:30 - 08:50 **"Opening Ceremony"** (Auditorium, Building No.8)

08:50 - 10:10 **Keynote-1** (Auditorium, Building No.8)

Chair: Kazunori SHIOYA (SRA-KTL, Tokyo)

Abstraction and Modelling in Formal Approaches to System Development

Prof. Keijiro Araki (Kyushu University, Japan)

10:10 - 10:30 Break

10:30 - 12:00 **Technical Sessions 1A and 1B** (Building No.1)

Session 1A: Engineering (1): Software Development Paradigm

Chair: SooYong Park (Sogang University, Seoul)

S1A-2: Creating and Maintaining Sustainable Open Source Software Communities

Yunwen Ye* (University of Colorad, USA)

Kouichi Kishida (SRA-KTL, Tokyo)

Kumiyo Nakakoji, Yasuhiro Yamamoto (University of Tokyo, Tokyo)

S1A-3: Support for Joining in Open Source Software Development Projects Using Public Data

Mitsunori FUJITA*, Kazuhiro Fujieda, Katsuhiko Gondow, Koichiro Ochimizu

(JAIST, Japan)

Session 1B: Internet (1): Security and Web-based Information Systems

Chair: Kazunori SHIOYA (SRA-KTL, Tokyo)

S1B-1: The Internet Key Exchange Protocol in IP Security

ZHU Jianming, SHI Tingjun, MA Jianfeng

(Xidian University, Xi'an and Chinese Academy of Sciences, Beijing)

12:30 - 14:00 Lunch

14:00 - 16:00 **Technical Sessions 2A and 2B** (Building No.1)

Session 2A: Engineering (2): Requirements, Product Line and Software Development Paradigm

Chair: Kei Kurakawa (NAIST, Nara)

S2A-1: A Conceptual Design Support System for Structuring and Visualizing Design Information Grounded by Scenarios

Kei Kurakawa*, Kumiyo Nakakoji (NAIST, Japan)

S2A-2: A Requirements Engineering Environment for Embedded Real-Time System -- SREE

Shu Fengdi*, Wu Guoqing (Wuhan University, Wuhan)

Tetsuo Tamai (University of Tokyo, Tokyo)

S2A-3: Feature Specification Method in Product-Line Development

Jaeseung Song, Minsung Kim, Sooyong Park* (Sogang University, Seoul)

Session 2B: Internet (2): Web-based Information Systems and Security

Chair: Yukinori GOTO/SAKAMOTO (Kyushu University, Fukuoka)

S2B-1: Design of an Aspect-Oriented Software Architecture for Web-based Information System

Atsushi Kumazaki*, Masami Noro, Han-Myung Chang, Yoshinari Hachisu

(Nanzan University, Japan)

S2B-3: UML Based Web Services Application Development Method

Jintae Kim, Eunjeong Kim*, Mansoo Seo, Sooyong Park (Sogang University, Korea)

S2B-4: Event Information Inputting and Publishing System developed by Social Science Students using XML & XSLT

Akira INOUE (Konan University, Kobe), Jun-ichi IGARI (Mainichi Newspapers, Tokyo), Ayako KOYAMA*, Shigeo KANEDA (Doshisha University, Kyoto)

16:00 - 16:30 Break

16:30 - 18:00 **Technical Sessions 3A and 3B** (Building No.1)

Session 3A: Distributed Computing and Mobile/Multi Agent Systems

Chair: Ju Dehua (ASTI, Shanghai)

S3A-1: Modeling and Verification of A Management System for TINA Network Object

O-Hoon Choi, Doo-Kwon Baik (Korea University, Seoul)

S3A-2: Atomic Commitment for Distributed Real-Time Transactions

YunSheng Liu, GuoQiong Liao (HUST, Wuhan)

S3A-3: Analyst-type Secretary Agent System for Schedule Management
Hiroshi Igaki*, Hajimu Iida, Ken'ichi Matsumoto (NAIST, Nara)
Makoto Sakai (SRA-KTL, Nara)

Session 3B: User Interface and Architecture

Chair: Kuo Yuesun (Academia Sinica, Taipei)

S3B-1: A Simplified Graph Model for User Interface Constraints
Chuan-chieh Jung (National Central University, Tao-yuan)
Tze-Heng Ma and Y. S. Kuo.(Academia Sinica, Taipei)

S3B-2: Metadata Based Virtual Schema Supporting Decision Making
JuHum Kweon, Doo-Kwon Baik (Korea University, Seoul)

S3B-3: A Decentralised Architecture for Workflow Support
Jun Yan, Yun Yang and Gitesh K. Raikundalia
(Swinburne University of Technology, Australia)

18:00 - 18:30 Break

18:30 - 20:00 **Reception**

Day 2: Oct. 24 (Thursday)

08:30 - 10:00 **Technical Sessions 4A and 4B** (Building No.1)

Session 4A: Formal Methods, System Evolution and Maintenance (1)

Chair: Syusaku IIDA (Sensyu University, Tokyo)

S4A-1: Support for Maintaining Distributed Component-based Systems
Xuefen Fang*, Tetsuo Tamai (University of Tokyo, Japan)

S4A-2: A Technique for Extracting EJB Components from Servlets
DONG KWAN KIM, YOUNG JONG YANG, HYO TAEG JUNG (ETRI, Daejon)

S4A-3: Modeling and Safety Analysis of Moving Block Railway Interlocking System
Nazir Ahmad Zafar* and Keijiro Araki (Kyushu University, Fukuoka)

Session 4B: Knowledge/Data Mining, Discovery and Engineering (1)

Chair: Miguel A. Serrano (CIMAT, MEXICO)

S4B-1: Digital Watermarking Algorithm Using Extraction Of Contours

Doo-Hyun Kim*, Yong-Sung Kim, OK-Chang , Cheol-Jung Yoo

(Chonbuk University, Korea)

S4B-2: Automatic Flowchart Layout for Program Visualization

Wei Lai (Swinburne University of Technology, Australia)

Mao Lin Huang* (University of Technology, Sydney)

S4B-3: Progressive Data Integration Based on Data Visibility

Dong Won Jeong, Doo-Kwon Baik* (Korea University, Korea)

10:00 - 10:30 Break

10:30 - 12:30 **Technical Sessions 5A and 5B** (Building No.1)

Session 5A: Formal Methods, System Evolution and Maintenance (2)

Chair: Jae Joon Lee (POSTECH, Korea)

S5A-1: Evolutionary Prototyping Technique using Abstract Interpretation in Java

Hiroyuki Ozaki*, Katsuhiko Gondow, Takuya Katayama (JAIST, Japan)

S5A-2: Specification Technique of EJB-Based Application Using Design By Contracts Approach

Hye-Min Noh*, Hyong-Jin Park, Cheol-Jung Yoo, Yong-Sung Kim and Ok-Bae Chang (Chonbuk University, Korea)

S5A-3: A Formal Refinement Method of Statechart

Shusaku Iida* (Senshu University, Tokyo)

Kokichi Futatsugi (JAIST, Japan)

Session 5B: Knowledge/Data Mining, Discovery and Engineering (2)

Chair: Kei Kurakawa (NAIST, Nara)

S5B-1: Commit Mechanism of Engineering Database Supporting Cooperative Design Transaction

Chen Guoning*, Taoshen Li, Guoqiong Liao (Guangxi University, Nanning)

S5B-2: Preserving Constraints in Mapping XML DTD to Relational Schema

Hao Zhong*, Yunsheng Liu, Yi Wang, Mingjun Chen (HUST, Wuhan)

S5B-3: A document categorization algorithm using the fuzzy set theory and hierarchical structure of documents

Hye-Jue Eun*, Seok-Woo Han, Doo-Hyun Kim, Yong-Sung Kim, OK-Chang, Cheol-Jung Yoo (Chonbuk University, Korea)

12:30 - 14:00 Lunch

Poster Session: **Tool Demonstration **CANCELED****

14:00 - 15:30 **Technical Paper Sessions 6A, 6B and 6C** (Building No.1)

Session 6A: Algorithms, Logics, Graph, Data Communication, Hardware (1)

Chair: Nazir Ahmad Zafar (Kyushu University, Fukuoka)

S6A-1: Improving Communication Quality with Reed Solomon Code in Internet Voice Broadcasting System

Shingo KASHIMA*, Yukinori Goto, Keijiro Araki (Kyushu University, Fukuoka)

Session 6B: Management: Process, Project, Testing and Quality (1)

Chair: Yoshitaka Matsumura (SRA-KTL, Tokyo)

S6B-1: AN EXPERIENCE REPORT ON USING THE TEAM SOFTWARE PROCESS

Miguel A. Serrano*, Carlos Montes de Oca (CIMAT, Mexico)

S6B-2: Modeling of Software Configuration Management Processes by using Petri nets

K. Mahesh Kumar*, Gargi Keeni (TCS, India)

S6B-3: On the Effectiveness of Genetic Operators in Adaptive Random Testing

Sebastian NG*, F.C. Kuo, R.G. Merkel, T.Y. Chen

(Swinburne University of Technology, Australia)

Session 6C: New Application Technologies and Domains (1)

Chair: Jiro Tanaka (Tsukuba Univ, Tsukuba)

S6C-1: Research of XML Wrapper System based on XML/CORBA (*presented as **P3C-5***)

Xiu-fen Fu*, Wang Jing, Sun Jianfang, Yang Junchao

(Guangdong University of Technology, Guangzhou)

Lu Yansheng (HUST, Wuhan)

S6C-2: Organise Class Libraries based on Functionality

Huilin Ye* (Newcastle University, Australia)

Hanchang Liu (Southern Cross University, Australia)

S6C-3: UML-based Component Design Techniques

Eun Sook Cho* (Dongduk Women's University, Korea)

Soo Dong Kim, Sung Yul Rhew (Soongsil University, Seoul)

Chul Hong Kim (ETRI, Daejon)

15:30 - 16:00 Break

16:00 - 17:30 **Technical Paper Sessions &A, 7B and 7C** (Building No.1)

Session 7A: Algorithms, Logics, Graph, Data Communication, Hardware (2)

Chair: Doo-Kwon Baik (Korea University, Seoul)

S7A-1: A Data Broadcast Scheduling Strategy for Real-time Read-only Transactions in A Mobile Environment

YANG Jincai, LIU Yunsheng, LI Guohui (HUST, Wuhan)

S7A-2: Some Design Issues for An Enterprise Geographic Information System

Wei Lai, Donggang Yu (Swinburne University of Technology, Australia)

Yongliang Zhong (Hypersoft, Austraila)

[Mao Lin Huang* (University of Technology, Sydney)]

Session 7B: Management: Process, Project, Testing and Quality (2)

Chair: Sebastian NG. (Swinburne University of Technology, Australia)

S7B-1: A systematic approach to plan the Inspection process

Ambrish K Srivastava, Gopesh Sharma and Gargi Keeni (TCS, India)

S7B-2: A Software Complexity Measurement Based on C++ Source Code for Reverse Engineering

Jong-Wan Kim*, Sung-Yul Rhew (Soongsil University, Korea), Sung-Eun Lee (Dong Seoul College, Korea)

S7B-3: An Automatically Generated Runtime Environment for Testing the Functionality of EJB Components

Jihyun Lee*, Gyu-Sang Shin (ETRI, Korea)

Session 7C: New Application Technologies and Domains (2)

Chair: Akira Kumagai (PFU, Tokyo)

S7C-1: Object Oriented Programming in Spreadsheet

Jong-Myung Choi, Chae-Woo Yoo (Soongsil University, Korea)

Sae-Hun Yeom (Dong Seoul College, Korea)

S7C-2: GIGA: Graphical Definition of Production Rules

Hiroaki Kameyama*, Kazuhisa Iizuka, Buntarou Shizuki and Jiro Tanaka

(University of Tsukuba, Japan)

S7C-3: E-commerce and ontology

Jianming Yong (University of Southern Queensland, Australia)
Jun Yan (Swinburne University of Technology, Australia)

Evening Free

Day 3: Oct. 25 (Friday)

08:40 - 10:20 **Keynote-2** (Auditorium, Building No.1)
Chair: Kazunori SHIOYA (SRA-KTL, Tokyo)
Requirements Triage:
"The Most Important Part of Software Engineering . . .
and the Most Ignored,"
Prof. Alan Davis (University of Colorado at Colorado Springs, USA)

10:20 - 10:30 Break

10:30 - 11:00 **"Closing Ceremony"** (Auditorium, Building No.1)
Chair: Akira Kumagai - *General Co-Chair*
Closing speech:
Kouichi Kishida (SEA/SRA-KTL, Japan) - *Symposium Secretary*
Kazunori SHIOYA (SRA-KTL, Japan) - *Program Co-Chair*
YunSheng Liu (HUST, China) - *General Co-Chair*
Yanxiang He (Wuhan University, China) - *Program Co-Chair*
[SooYong Park (Sogang University, Korea) - *Program Co-Chair* (left)]

12:00 - 14:00 Lunch
Afternoon **Wuhan city tour** *or* official **Excursion to Three Gorges** (2 nights, 3 days)
Evening Free

***NOTE:** Finally, **41+1 papers were presented** (out of 49 papers were accepted), and **not presented papers are omitted from this program and the proceedings**.*

**: speaker name reported by the Session Chairs for note.*

***The poster program** is extracted from locally provided "program leaflet" and attached separately. Some are listed **in Chinesé**.*

This page is intentionally made blank

Summaries of presented papers

This page is intentionally made blank

Creating and Maintaining Sustainable Open Source Software Communities

Yunwen Ye[1,2] **Kouichi Kishida**[2] **Kumiyo Nakakoji**[3] **Yasuhiro Yamamoto**[3]

[1]Department of Computer Science
University of Colorado
Boulder, CO80309-0430, USA
yunwen@cs.colorado.edu

[2]SRA Key Technology Lab
3-12 Yotsuya, Shinjuku
Tokyo 1600-0004, Japan
k2@sra.co.jp

[3]PRESTO, JST, and
Knowledge Interaction Design Lab
RCAST, University of Tokyo
kumiyo@ai.rcast.u-tokyo.ac.jp *yxy@acm.org*

ABSTRACT

Open Source Software (OSS) systems are developed by volunteers, who form communities of practice due to their common interest in the systems. Understanding the motivation of the developers who volunteer to participate is of crucial importance to the sustainable development of OSS communities and hence OSS systems. In this paper, based on the learning theory of *Legitimate Peripheral Participation*, we argue that learning entailed by community participation is one motivating force for OSS developers. Our argument is supported through the analysis of the social structure of OSS communities and the co-evolution between OSS systems and communities.

INTRODUCTION

Open Source Software (OSS) systems are developed by volunteers. However, there is very limited, if any, understanding of what motivated so many software developers to dedicate their time, skills, and knowledge to OSS systems with no monetary benefits. OSS developers, users, and user-turned-developers form a community of practice. A *community of practice* is a group of people who are informally bounded by their common interest and practice. We argue that learning is one of the motivating forces that attract developers to participate in OSS development and to become members of OSS communities. The argument is grounded in the learning theory *Legitimate Peripheral Participation* (LPP).

OSS COMMUNITIES

The right to access and modify source code itself does not make OSS projects different from most "Closed Source Software" ones. The fundamental difference is the *role transformation* of the people involved in a project. In Closed Source Software, developers and users are clearly defined and strictly separated. In OSS, clear distinction between developers and users does not exist: All users are potential developers.

The distinct feature of *role transformation* in OSS projects leads to a different social structure. People involved in a particular OSS project create a community of practice. Members of an OSS community assume certain roles by themselves according to their personal interest in and contributions to the project. Our previous research studying four different OSS projects has found that a member may have one of the following eight roles: Passive User, Reader, Bug Reporter, Bug Fixer, Peripheral Developer, Active Developer, Core Member, and Project Leader.

The influences that members have on the system and the community are different depending on the roles they play. The role closer to the center has a larger influence. A Project Leader has a larger influence than that of a Core Member, who in turn has a larger influence than an Active Developer, and so on.

OSS systems and communities co-evolve. The evolution of an OSS community is effected by the contributions made by its aspiring and motivated members. Such contributions not only transform the role and influence of their contributors in the community and thus evolve the community, but lead to system evolution. The opposite is also true. Any improvement or extension made to an OSS system—whether it is a bug fix, a bug report, or a patch—not only evolves the system but also redefines the role of the contributing member and thus changes the social dynamics of the OSS community (Figure 1).

LEARNING AS THE MOTIVATIONAL FORCE

Factors that affect motivation are both intrinsic (cognitive) and extrinsic (social). The precondition for motivating developers to get involved in OSS is that they must derive an intrinsic satisfaction in their involvement in OSS. Intrinsic motivation is positively reinforced and amplified when social structure and conventions of the community recognize and reward the contributions of its members.

We argue that learning is one of the driving forces that motivate developers to get involved in OSS projects because learning provides the intrinsic satisfaction for OSS developers. We explain our theory in the framework of *Legitimate Peripheral Participation* (LPP) developed by Lave and Wenger based on their studies of communities of practice, which include OSS communities.

The Legitimate Peripheral Participation Theory

In the theory of LPP, learning is not viewed as a process of

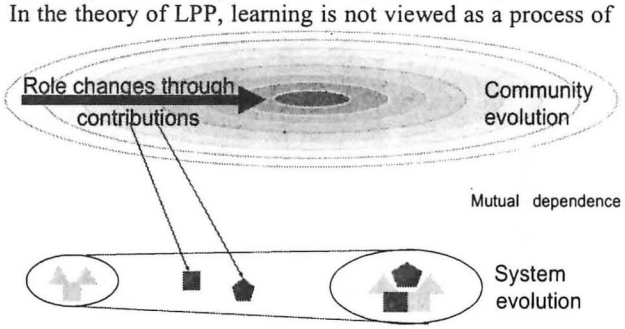


Figure 1: Co-evolution of systems and communities in OSS

gaining a discrete body of abstract knowledge that learners will then transport and reapply in later contexts; instead, learning is viewed as a process of acquiring skills through social interactions with other practitioners of the community, of transforming identities from a journeyman to a master in the community, and of changing relationships with other members of the community. When new comers enter a community as journeymen, they are given the *legitimate* access to the expertise of masters. At first, new comers learn by *peripherally participate* in small and easy tasks of actual practice with masters. As they become more competent and undertake more difficult tasks, they move gradually toward the center of the community and eventually establish their identities as masters.

LPP refers both to the identity development of individual members from journeymen to masters through a series of role transformations, and to the reproduction and evolution of the community. For a community to be sustainable over time, it must be able to reproduce itself through the transformation of roles and identities of its members.

Learning As the Intrinsic Motivation

To understand the role that learning plays in forming and sustaining OSS communities, we need to analyze its impact on the two types of OSS participants: those who initiate a new OSS project and those who join existing OSS projects.

Learning Experience of OSS Initiators

Initiators of OSS projects may be motivated by *explorative learning* and *learning by doing*. *Explorative learning* is similar to most scientific research in which learners (e.g., scientists, practitioners) attempt to find new ways of doing things or of overcoming an existing problem. In *learning by doing*, learners want to deepen their understanding of a certain domain by actually engaging in practical tasks that allow them to apply their existing knowledge and to perfect their current skills.

Learning by Participation

When the results of the above, more often than not, individual learning efforts are distributed in the form of open source, they provide resources and opportunities for other developers to learn. Most developers who start an OSS project are master programmers, and their systems are the products of fine craftsmanship and examples of excellent programming practice. When those systems are freely distributed, they grant other developers *legitimate* access to the embedded skills and knowledge. Similar to the way that we learn to write by reading literature, reading existing source code of expert programmers is a powerful path to the mastery of programming art.

The learning experience of later participants of OSS projects does not stop at passive absorption by reading source code; it also happens when new participants engage in bug reports, maintenance, and further development of OSS projects. In most cases, new participants do not become Core Members suddenly. As we have analyzed above, they have to earn their status and recognition in the community gradually by making small contributions at first. In other words, they start with *peripheral participation* by, for example, reporting and fixing bugs. By doing so, they learn by doing and their skills improve. As their skills are gradually recognized in the community based on their contributions, they are trusted to more challenging tasks, and move toward the "inner circle" of the community, executing larger influences.

Active participation of new members creates opportunities for them to interact with other more knowledgeably skilled developers, and gives them *legitimate* access to the expertise therein.

Social Aspects of the Extrinsic Motivation

The social fabric inherent in OSS communities reinforces the intrinsic motivation for participating in OSS projects as a form of learning. Only in a society where technical supremacy is highly appreciated can developers acquire good reputations among their peers by displaying their skills through the free distribution, and often wider acceptance, of their systems. The good reputation attracts attention, trust, and cooperation from others and lays the foundation for advancing the original developers' agenda and the establishment and development of OSS communities.

Members close to the center of the community enjoy better visibility and larger influences than do peripheral members. The road to the core has to be paved by contributing more to the project and interacting more with other members (Figure 1). As new members contribute to the system and the community, they are rewarded with higher recognition and trust, and higher influence as well.

An Oriental Perspective

The viewpoint of learning as motivation that intrinsically drives people to get involved in OSS development and that extrinsically rewards them with higher social status and larger influence in OSS communities is in parallel with a tradition of Eastern culture.

DISCUSSIONS

Realizing that learning is one of the major driving forces for OSS development has several practical implications.

First, developers at the center of OSS communities should pay more attention to the creation and maintenance of a self-reproducing OSS community. Second, the existence of many OSS projects provides a possibility for educators to change the way of educating and training software professionals in schools. Third, given the importance of learning in OSS communities, the importance of the skill of reading programs and tools that support program reading and understanding should be more stressed.

Support for Joining in Open Source Software Development Projects Using Public Data

Mitsunori Fujita Kazuhiro Fujieda Katsuhiko Gondow Koichiro Ochimizu
School of Information Science, Japan Advanced Institute of Science and Technology
1-1 Asahidai, Tatsunokuchi, Nomi, Ishikawa 923-1292, JAPAN {fjt, fujieda, gondow, ochimizu}@jaist.ac.jp

ABSTRACT

In Open Source Software Development (OSSD), many superior developers develop and test the software rapidly in parallel and distributed manner. This development process, however, makes newcomers hard to join in a project because they need to know various information about the project. The newcomer has to spend the time and effort for reading many source codes, documents and archives of mailing lists. To contribute his changes, he also has to learn the programming technique. In this paper, we propose a new method of supporting newcomer to get useful information generated from the public data. The public data is provided by tools generally used in OSSD projects without adding extra restrictions to a development style.

INTRODUCTION

The OSSD is widely acknowledged as a useful way of software evolution. In OSSD, the process of development, debugging and testing are performed repeatedly in parallel by the core members. They perform configuration management of each module of software with receiving the problem report, the idea of improving software, and the source code of enabling new features from users. To evolve the software, continuous and quick improvement cycle is required. The cycle makes a user getting interested in OSSD a developer. Not all users, however, who have trouble in software, could not contribute to the project. We consider the following two problems prevent newcomer from joining in OSSD:

1. Difficulty of grasping the progress of OSSD
2. Difficulty of learning programming technique

In this paper, we propose two methods that support users joining in OSSD to solving these problems.

APPROACH

The goal of our research is to solve the problems by providing some tools which give us useful information extracted from public data without adding any change of the general development style of OSSD. The public data is provided by systems generally used in OSSD. The systems include version control systems, mailing lists and bug tracking systems. To solve the problems and evaluate the usefulness of our approach, we have developed methods and tools that help us extract information to be considered useful, such as access logs and a history of changes, from the public data.

THE METHOD OF GRASPING THE STRUCTURE OF OSS, AND PROGRESS OF DEVELOPMENT

To solve the problem 1, we provide the information which combines relevant data extracted from public data. We call this information "Project information", which includes necessary information to know about project. The information includes the latest procedure of development and contribution to join in OSSD, a structure of software to modify source codes, a module and file owner to discuss about it, etc. It is difficult to know the project information directly because it is mixed in many messages of mailing list, documents and source codes.

Our approach is to obtain various information from public data and to analyze it to extract the project information. For example, we can find the change request and announcements about it by keyword searching and check the date of messages from archives of mailing list to get the project information. Then we can know the change in the period about the request by using logs of version control system. This change includes the filenames updated simultaneously and the names of developers who modified these files. By checking a change history, we assume the files changed simultaneously have strong relation and the developer who frequently changes a file is an owner of the file. We can verify the validity of the project information by the investigation of the dependency of files. We can also read mailing list archives or post the question to a developer

mailing list.

CVSLogMonitor

We have developed the tool "CVSLogMonitor". It extracts and shows the change history, the update frequency and the update relevancy by analyzing outputs of a CVS command "*log*". CVSLogMonitor supports us to extract such project information.

Example of data extraction

We analyzed project information in some OSS development project to ensure our approach. An example is the result of analysis about the GIMP project. There are three steps to analyze project information.

1. To search files related to the addition of a new function.
2. To specify files that have strong relation to the files.
3. To specify developer who is mainly maintaining the file.

To take these steps, we specified the changes about adding new function, the relevant files and the owner of the files.

We think these project information extracted from the public data shows the possibility and the usefulness of our approach.

THE METHOD OF TEACHING / LEARNING PROGRAMMING TECHNIQUE

To solve the Problem 2, we prepared exercises extracted both from messages of a problem report and the changes which have been applied to software in order for users to learn advanced programming techniques. It is necessary for programmers to train to develop OSS cooperatively, because there are very few guidebooks for debugging and maintaining software.

The Concept of Exercises

The public data includes many problem reports and modifications of the problems. The problem reports show bugs actually occurred in the software. The modifications include elegant source codes refined by developers. We thought that the problem reports and the modifications are good example to learn programming technique. Our idea is that we use a bug which has been actually reported to the project as a precious source of the exercises of debugging. The exercise consists of a problem report, a source tree of the software including bugs and an example of an answer.

Preliminary experiment for evaluating exercises

To evaluate our exercises, we have made plan to do an experiment. In the current state of the practice, we performed the preliminary experiment.

In the experiment, a testee tried to solve an exercise. We checked differences between codes and took interview with him about the exercise when he finished solving it. After having a testee solve the exercise problem, we passed an example of an answer and performed the code review. Although problem creation has high cost, the exercise is impressive for the testee. The exercise motivates him in the preliminary experiment. We perform an experiment to detail evaluation of the obtained result. In addition, we research the method and the tool to generate exercises semi-automatically, because the current method depends heavily on the hand works.

RELATED WORKS

There are some researches on the OSSD using public data generated by CVS, mailing list and bug tracking system. Their researches have only described the process and the actual condition of OSSD objectively. We directly aim at support of OSSD by providing methods and tools.

CONCLUSION

In this paper, we described our approach to support newcomers to join in OSSD using public data. The information extracted from public data is effective for understanding a project and for creating the exercise problems which enhance subject's motivation to solve. Our approach is suitable for supporting OSSD, to decrease the time and effort of reading many source codes and mailing lists without adding restrictions to a development style of an open source. A detail analysis on the validity by an experiment is future work.

The Internet Key Exchange Protocol in IP Security

ZHU Jianming SHI Tingjun MA Jianfeng
School of Computer, Xidian University, Xi'an 710071 China
State Key Laboratory for Information Security, Chinese Academy of Sciences, Beijing 100039 China
jmzhu@yeah.net

ABSTRACT

Internet Key Exchange (IKE) is an automated key exchange mechanism that is used to facilitate the transfer of IPSec security associations (SAs). Public Key Infrastructure (PKI) are being considered as a key element for providing security to new distributed communication networks and services. In this paper, we concentrate on the properties of the protocol of Phase 1 IKE. After investigating IKE protocol and PKI technology, we combine IKE protocol and PKI and present an implementation scheme of the IKE based on PKI. Then, we give a logic analysis of the proposed protocol with the BAN-logic and discuss the security of the protocol. The result indicates that the protocol is correct and satisfies the security requirements of Internet key exchange.

KEYWORDS

Authentication, encryption, IKE, PKI

INTRODUCTION

With the development of Internet, the security of IP is of increasing important. As one of the most important technologies of IPSec, Internet Key Exchange (IKE) has received significant interest from the research community. IKE is the default automated key management protocol selected for using with Internet Protocol Security protocol (IPSec).

The IKE protocol has two phases: in the first phase a secure channel between the two key management daemons can be establishes, while in the second IPSec SAs can be directly negotiated. RFC 2409 defines four different authentication methods for Phase I protocols: preshared key, public key signature, public key encryption, and revised public key encryption. In this article, we focus on the authentication and key generation mechanisms of IKE using public key, then we propose an authentication method based on public key Infrastructure (PKI).

A SCHEME OF IKE BASED ON PKI

We chose to develop a public key infrastructure (PKI) for managing X.509 v.3 public key certificates and v.2 certificate revocation lists (CRLs) issued to Internet nodes.

A certificate defined in X.509 contains the user's public key and other information and a signature of these information by CA (Certificate Authority). For example, the equations below are certificates.

$CERT_I=\{ID_I,KU_I,Date_I,LF_I, E_{KR_{CA}}(ID_I,KU_I,Date_I,LF_I)\}$

$CERT_R=\{ID_R,KU_R,Date_R,LF_R, E_{KR_{CA}}(ID_R,KU_R,Date_R,LF_R)\}$

$CERT_{CA}=\{ID_{CA}, KU_{CA}, Date_{CA}, LF_{CA}, E_{KR_{TCA}}(ID_{CA}, KU_{CA}, Date_{CA}, LF_{CA})\}$

Where $CERT_X$ represents the certificate of X, in which ID_X means the identity of entity X, KU_X is the public key of entity X, KR_X is the private key of entity X, $Date_X$ is the issue date of the certificate to X, and LF_X is the lifetime. These data are signed by CA using its private key KR_{CA} and the data in $CERT_{CA}$ are signed by the top level certification authority (TLCA) using its private key KR_{TLCA}. $E_X(Y)$ indicates that "Y" is encrypted with the asymmetry key "X". $(Y)_X$ indicates that "Y" is encrypted with the symmetry key "X".

We propose the PKI-based IKE protocol shown in Figure 1.

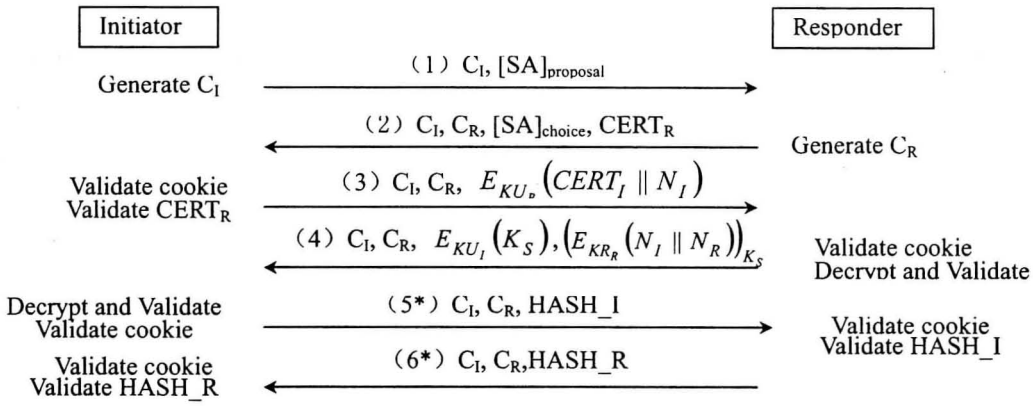


Figure 1. Secret Key negotiation based PKI

Step 1: The initiator generates a random number, cookie C_I and proposes SA.

Step 2: The responder generates a request for confirmation, $<C_I, C_R>$, chooses one and only one proposal from the list in $[SA]_{proposal}$. Then, he sends C_I, C_R, $[SA]_{choice}$ and $CERT_R$ to the supposed initiator.

Step 3: After receiving the message, Initiator should validate cookie C_R and certificate $CERT_R$ using the public key of CA. If they are not valid, Initiator terminates the execution. Otherwise, Initiator generates a nonce N_I and uses the public key of Responder to encrypt the message ($CERT_I$|| N_I), where "||" means concatenation. Then, Initiator sends message (3) to Responder.

Step 4: After receiving message (3) from Initiator, Responder decrypts it to get $CERT_I$ and N_I. Then, Responder verifies the validity of certificate $CERT_I$ using the public key of CA. If it is not valid, Responder terminates the execution. Otherwise, Responder generates a session key K_S and a nonce N_R, using its private key KR_R to sign ($N_I||N_R$), the public key of Initiator to encrypt K_S, and K_S to encrypt the message ($E_{KR_R}(N_I \| N_R)$). Finally, Responder sends these messages to Initiator.

Step 5: After receiving message (4) from Responder, Initiator opens the message, decrypts it to get K_S and ($E_{KR_R}(N_I \| N_R)$) and recovers ($N_I||N_R$) using the public key of Responder. Then, Initiator generates HASH_I and sends it to Responder, where

$$HASH\_I = prf(SKEYID, K_S|| C_I || C_R || SA || ID_I)$$

Step 6: After receiving message (5) from Initiator, Responder validates HASH_I. If it is valid, Responder generates HASH_R and send it to Initiator, where

$$HASH\_R = prf(SKEYID, K_S || C_R || C_I || SA || ID_R)$$

"SA" in HASH_I and HASH_R is the SA payload sent by the Initiator. "Prf" is a pseudorandom function usually implemented by a keyed-hash such as HMAC; the exact mathematical transformation is determined during the parameter negotiation. SKEYID is the key to prf, it is derived from the nonces:

$$SKEYID = prf(hash(N_I||N_R), C_I||C_R)$$

where "hash" is a hash algorithm determined during the parameter negotiation. Since the nonces are shared secrets, the SKEYID is also a shared secret and fresh; therefore, HASH_I and HASH_R can be used directly for authentication.

CONCLUSION

In this paper, we propose a scheme of the Phase I protocols of IKE based on PKI. The proposed protocol is analyzed formally with BAN-logic and proves its correctness. By discussing the security of the protocol, we point out that our protocol satisfies the security requirements of Internet key exchange. With the development of PKI, the proposed protocol is practical.

A conceptual design support system for structuring and visualizing design information grounded by scenarios

Kei Kurakawa
Nara Institute of Science and Technology, Japan
8916-5 Takayama, Ikoma, Nara, 630-0101, JAPAN
kurakawa@is.aist-nara.ac.jp

Kumiyo Nakakoji
RCAST, The University of Tokyo
4-6-1 Komaba, Meguro, Tokyo, 153-8904, JAPAN
kumiyo@kid.rcast.u-tokyo.ac.jp

KEYWORDS

Conceptual design, Computational support, Knowledge representation, Scenario, Design rationale

ABSTRACT

One of the challenging issues that designers face during a conceptual design phase is to maintain a variety of complex information, which emerges during the conceptual design task. This paper presents our approach to support designers in understanding such information including design rationale during and after the conceptual process. We propose the CDS (Conceptual Design guided by Scenarios) model (Figure 1, 2, 3), and present a design method that helps designers externalize and structure design information based on the model (Figure 4, 5). We describe the CD-Scenery system (Figure 6), which we have developed based on the method, and present a user study conducted in order to validate the underlying method.

The CDS (Conceptual Desgin guided by Scenarios) model

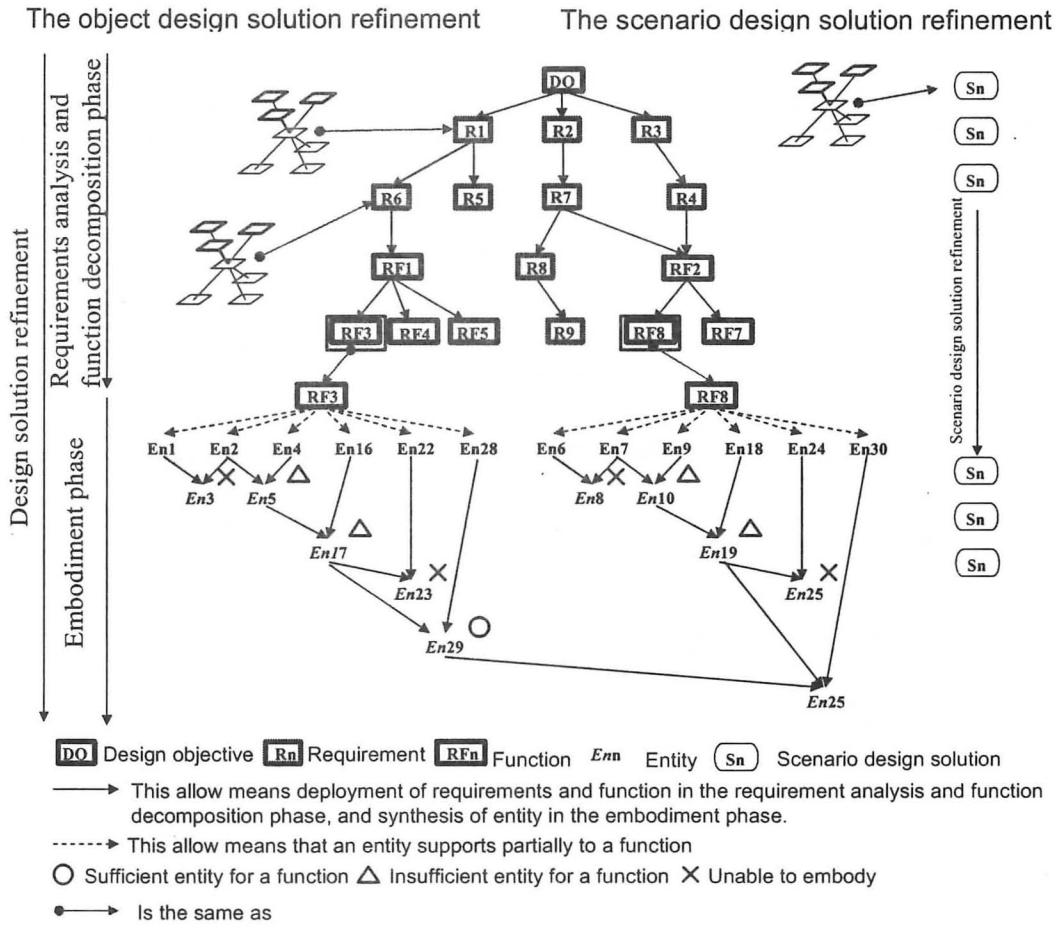


Figure 1 The design process grounded by scenarios

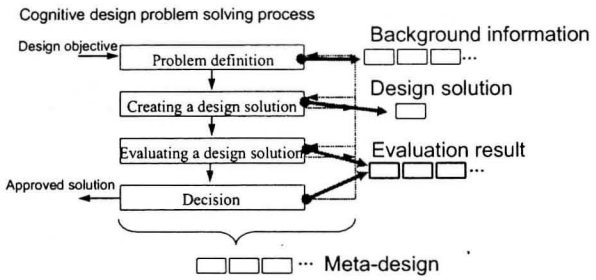


Figure 2 A cognitive design problem solving process and types of design information

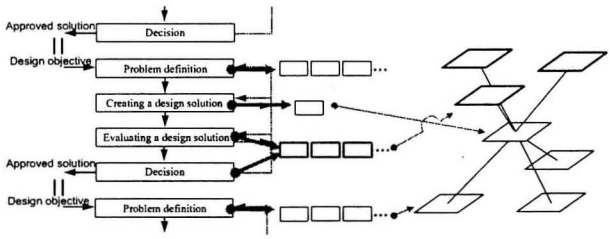


Figure 3 A design information structure

The method and its application

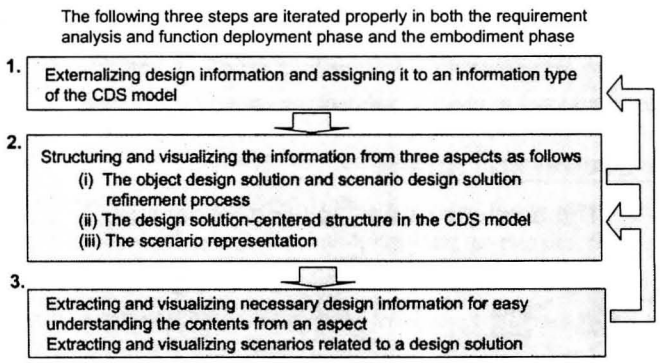


Figure 4 The design procedure based on the CDS model

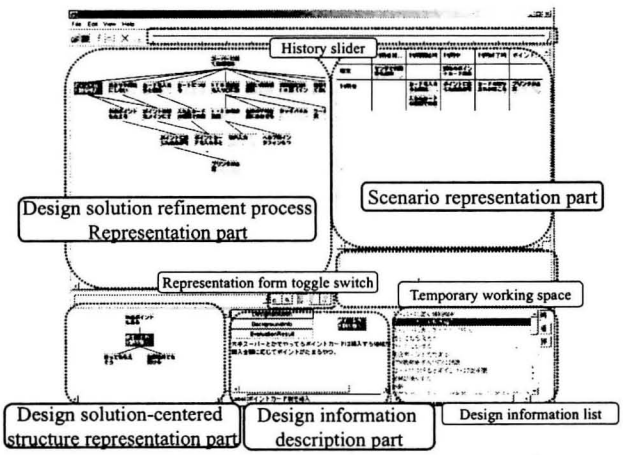


Figure 6 A sample image of the CD-Scenery

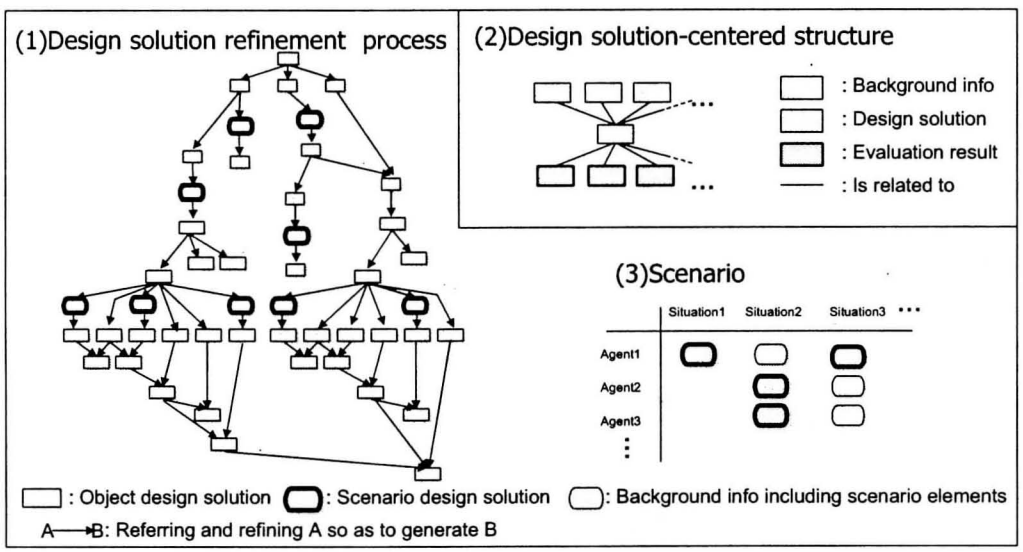


Figure 5 Three aspects and their representation

A Requirements Engineering Environment for Embedded Real-Time System — SREE

Shu Fengdi **Wu Guoqing**
Department of Computer Science
Wuhan University
Wuhan 430072, P.R.China
fdshu2002@yahoo.com.cn wgq@whu.edu.cn

Tetsuo Tamai
Interfaculty Initiative for Information Studies
Graduate School of Interdisciplinary Information Studies
University of Tokyo, Japan
tamai@graco.c.u-tokyo.ac.jp

Due to the characteristics of Embedded real-time software (ERSW for short), it's a great challenge to describe its requirements completely and accurately. This paper introduces SREE, a requirements engineering environment developed by us which can support most tasks in requirements phase. SREE specifies requirements of ERS with templates and formal rules, and applies prototyping method and visual techniques to requirements verification.

THE REQUIREMENTS SPECIFICATION MODEL AND DYNAMIC EXECUTION MODEL

RTRSM is a requirements specification model based on hierarchical finite state machine (HFSM for short) similar to Statecharts. It represents HFSM with templates and rules, where one template corresponds to one finite state machine (FSM for short) and one rule corresponds to one transition in FSM. Rules are of the form:

s_r, e (att(e)) $\Rightarrow$ s_{r+1} When Cond Do EF

To represent the parallel relationship of substates of an AND-state, Decomposing rule and Terminating decomposition rule are defined. Specification based on RTRSM consists of a collection of templates, which are described with forms. Each template comprises information with respect to a FSM or subsystem. Rules are the formal representation of FSM, and we can simulate the dynamic behaviors of ERS by executing these rules, which results in the executability of RTRSM. To simulate the execution of ERS, RTRSM provides the following functions with respect to time constraints and system clock: GetTime, SetTimer, and ClearTimer. Besides, RTRSM provides function SendControl to describe how ERSW sends commands to the external environment and function SendNotify to broadcast generated internal events. The paper presents the definition of state configuration and describes the dynamic execution model based on control flow and data flow (DEM for short) of a complex ERS by a 8-tuple (GC, C_0, GT, θ, V, GD, GU, δ). Successively, the execution sequence based on DEM is described by a six-tuple(C, gt, gd, gu, time, C').

OVERVIEW OF SREE

The rule generator automatically translates input graphic symbols into rules related to corresponding templates, and the specification editor also is responsible for syntax checking.

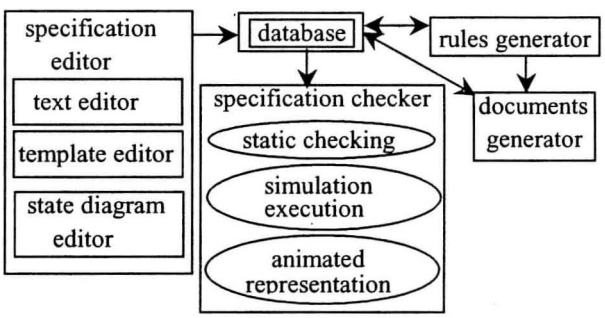


Fig. 1. The configuration of SREE.

Properties to be checked should consist of consistency, completeness and security of SRS. In SREE, the analysis and checking of SRS are based on five elements: state, transition, guarding condition, event and timestamp. It

includes static analysis and checking, simulation execution, which is available after syntax checking, and animated representation. The first includes single step and single path checking, reachability checking, and consistency checking. And the later two are based on the working model of ERS. The user can input one or several stimulus events with their attribute values and check SRS in interactive mode step by step or in batch with input file. In checking process, the controller of simulation execution can display checking information on the screen, save some executing results to the output file and send commands to the animation demo system. SREE makes use of visual techniques mainly based on animated representation to show the working process of ERS for users. It provides a general interface with which to communicate with animation demo systems, which can be taken as the simulation of the practical external environment of ERSW. When the specification checker receives an external event from the interface, it executes corresponding processes, shows changes of states in the state diagram and sends control commands to the animation demo system through the interface. On accepting these commands, the animation demo system shows corresponding animation scenarios. With the cooperation of the controller of simulation execution and animation demo systems, SREE enables users to see, in the same screen, the state diagram show changes of states with color marks while the animation demo system display corresponding scenarios.

CONCLUSIONS

This paper has introduced SREE that is designed for the requirements specification and verification of ERS. RTRSM only builds one state-based model and is an executable language, representing requirements with formal rules and templates. Graphic symbols can be translated into rules automatically as its underlying formal representation, and information to be filled in templates is simple. RTRSM satisfies most design criteria of specification language, such as minimality, simplicity and conciseness. Due to the abstract characteristics of HFSM and that abstract demonstrations or detailed program modules can represent the execution of actions in rules, RTRSM can support different levels of abstraction and refinement. Furthermore, SREE can not only provide tools for specification editing and analysis as many other working environments, but support specification-based prototyping method and provide interface connected with animated graphics, by which users can review the working overview of target systems in requirements phase. Additionally, the theoretical basis of the dynamic specification checking method of SREE is DEM, which is based on control flow and data flow. What's more, SREE can combine with technology of software components. It is our main goals to study more effective and applicable model and method to specify and verify requirements of ERS, and probe how to combine formal methods with visual techniques efficiently. Our future work is to polish SREE and apply it to more complex and larger embedded real-time systems.

Feature Specification Method in Product-Line Development

Jaeseung Song **Minsung Kim** **Sooyong Park** **Keunghae Lee**
Department of Computer Science and Engineering
Sogang University
Sinsu-Dong, Mapo-Gu, Seoul, 121-742, REP. of KOREA
{anthony, mskim}@selab.sogang.ac.kr sypark@ccs.sogang.ac.kr khlee@hanul.hangkong.ac.kr

SUMMARY

1. INTRODUCTION

Software product line is one of the most promising approaches to increasing reusability based on commonality across related software systems. Research of Product Line Development, which is used to improve reusability among product families, has progressed actively because of the diversity of customer's needs, rapidly changing market requirements and the quick response of market competition. Feature Oriented Domain Analysis (FODA) is commonly used in Product Line Development to analyze commonalities and differences among a family of products in terms of product features.

However, there have been many problems because many parts in feature modeling are dependent on the experiences of a domain expert. A feature model in today's feature-oriented approach is specified informally. The research on formal specification for features is insufficient recently.

In this paper, we propose a solution to the above problems by formalizing features by using a multi-paradigm formal specification language. Feature specification processes are described to specify features formally. Also, feature interaction management is used to verify suggested specification method and the proposed feature specification method is applied to Distributed Meeting Scheduler System domain.

2. RELATED WORK AND THE PROBLEMS OF FEATURE MODEL

2.1 FODA in Product-Line Development

Since FODA was introduced in 1990 by the Software Engineering Institute, many domain engineering and product line engineering methods have adopted the technique to support feature-oriented commonality and variability analysis.

2.2 Formal Specification

Formal specification is the expression of a collection of properties some systems must satisfy in some formal language and at some level of abstraction.

2.3 The Problems of Existing Feature Model

In a feature-oriented analysis, like FODA, feature modeling is dependent on the experiences of a domain expert and features are specified informally. Informal specification may cause problems, such as ambiguities, interpretation errors, incompleteness and so on.

3. FORMAL SPECIFICATION AND VERIFYING METHOD FOR FEATURE MODEL

Following three things are proposed to specify feature formally. First, The structure and language of feature model are defined at meta-level. A feature model is specified using multi paradigm language. Second, Feature Specification Process is suggested to define all the process from feature acquisition to specification. Finally, Feature Interaction Management is used to verify proposed specification method.

3.1 Specification Method for Feature Model

A Meta-Level Feature Model Using UML

A feature should be abstracted and modeled at the meta-level to describe domain independent language and structure.

Structure and Formal Language of the Feature

we propose how to specify a feature using sets of predicates obtained from meta-level feature model. Figure 1 presents the structure of a feature using UML.

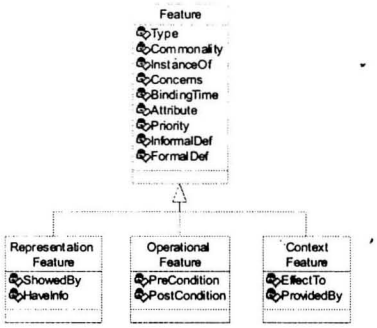


Figure 1 Structure Diagram of Feature using UML

Feature Specification using Formal Method

Formal specification of feature should be accomplished to resolve problems caused from informal specification. Z and Temporal Logic are used as formal specification languages of feature to describe both functional and non-functional attributes.

3.2 Feature Specification Process

Feature Specification Process is composed into following five main steps.

i. Initiate Feature Identification : \
ii. Meta-Level Feature Model :
iii. Formalize Feature
iv. Propagation
v. Formally Specify Feature Model

3.3 Verifying Feature Specification Method

Feature Interaction Management is used to verify proposed feature specification methods.

Definition of Feature Interaction Management

Interaction Management among features are defined in this paper as follows:

Feature Interaction Management is the set of activities concerned with the identification, categorization, resolution and management of critical relationships among sets of feature used to describe a system.

4. CASE STUDY : DISTRIBUTED MEETING SCHEDULER DOMAIN

In this section, we apply our feature specification method to the Distributed Meeting Scheduler System domain. Problems caused by informal specification of feature model are analyzed in previous section. The problems are resolved through formal specification as demonstrated in the suggested feature specification process.

5. CONCLUSION AND FUTURE WORKS

In this paper, formal specification method of feature, which is used to distinguish commonalities and variabilities between product families, are proposed to resolve problems caused by informal specification. Interpretation Error, Ambiguity, Incompleteness and Interaction Problems caused from informal specification of feature are resolved through formal feature specification methods proposed.

This paper has proposed a meta-level feature model for defining domain-independent model and predicates used in formal language. Feature specification processes have been proposed to describe how to specify features formally. Feature interaction management has been used to verify the suggested specification methods. And the proposed feature specification method has been applied to Distributed Meeting Scheduler System domain.

Formal Specification Method and Process proposed here can be seen as a preliminary attempt to reason more formally in terms of informal feature.

For the future work, we will continue to research in the areas of extended specification method including business strategy aspects issued in product line development. Finally, future research should be applied to other domains to obtain new experiences on feature specification.

Design of an Aspect-Oriented Software Architecture for Web-based Information System

Atsushi Kumazaki

Dept. Info. Sys. & Quant. Sci., Nanzan Univ.
18 Yamazato, Showa, Nagoya, Japan 466-8673
d00bb002@nanzan-u.ac.jp

Masami Noro, Han-Myung Chang
Yoshinari Hachisu
Dept. Info. & Telecomm. Eng., Nanzan Univ.
27 Seirei, Seto, Japan 489-0863
{yoshie, chang, hachisu}@nanzan-u.ac.jp

1 Introduction

This paper describes an aspect-oriented software architecture for Web-based information systems(WIS), which is aspect-oriented version of the architecture we previously constructed. The key issues for developing WIS, in these days, are how to select and coordinate existing Web-based technologies such as CGI scripts, applets, PHPs, servlets and so on. There is unfortunately a bit chaos and confusion on the coordination and they enforce troublesome development. An aspect-oriented software architecture implies a concurrent software process. The architecture is a guide map for the selection and the coordination. It eases the development of WIS as a result.

2 Software Architecture

We realized that a software architecture is not just a product model but also a container including a process planning scenario. The insight we gained meets the definition by the SEI discussion group in '94 of a software architecture that "the structure of the components of a program/system, their interrelationships, and principles and guidelines governing their design and evolution over time."

The software architecture includes components from the software model, the implementation stage, and the design result. Fig.1 shows our software architecture including a process planning scenario.

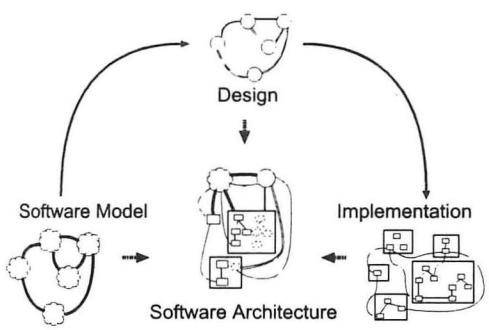
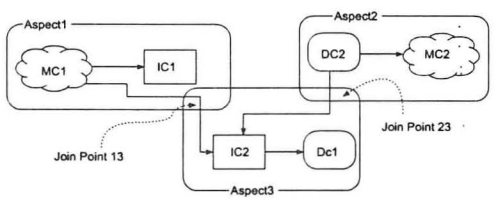


Fig. 1: Software Architecture

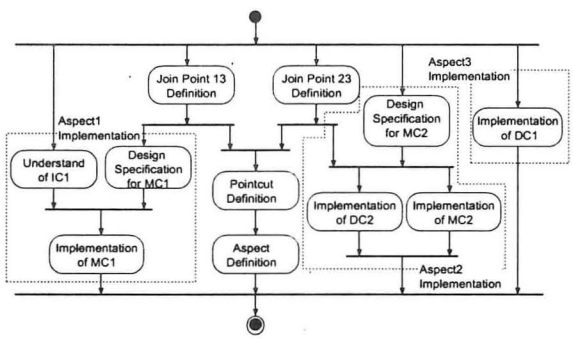
2.1 Aspect, Architecture and Process

We consider that an aspect-oriented software architecture is composed of a set of aspects. An aspect is a composite component. In this sense, if we apply a component-connector model to an aspect-oriented software architecture, an aspect is a component, and a join point is a connector.

In an aspect-oriented software architecture, implementation processes for the development of aspects are combined into a implementation process for the architecture. Definitions for inter-aspect protocols (join points) precede the processes for the aspects. If a software architecture consists of three aspects crosscutting one another and each aspect is composed of components as in Fig.2 (a), the implementation process will be one shown in (b).



(a) an Aspect-Oriented Software Architecture



(b) Software Process Implied by the Architecture

Fig. 2: Aspect-Oriented Software Architecture and its Implementation Process

3 Aspect-Oriented Software Architecture for WIS

To represent a implementation process in a software architecture, we describe a software architecture from the following three views: abstract view, concrete view and process view.

3.1 Aspects in WIS

Here, we describe aspects in WIS. The following concerns are recognized through observing the construction of the software architecture for WIS : Controller, View, Application Logic, Platform, Page, Efficiency and Security. Fig.3 shows these concerns, aspects, and their relationships in a WIS.

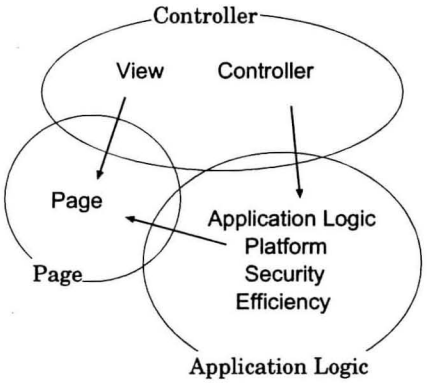


Fig. 3: Concerns, Aspects, and their Relationships in a WIS

3.2 Abstract View of WIS Architecture

Our aspect-oriented software architecture for WIS is aspect-oriented version of the architecture we previously constructed. We borrow portions of the previous architecture to draw the aspect-oriented architecture. Fig.4 outlines our aspect-oriented architecture for WIS.

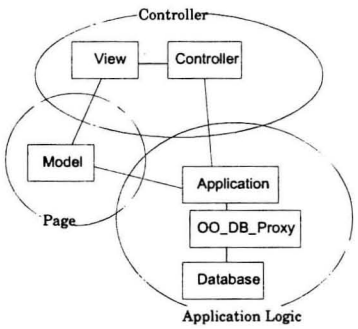


Fig. 4: Abstract View

3.3 Concrete View of WIS Architecture

The concrete view defines abstraction levels of components. The application logic aspect includes application dependent objects. Since they are heavily application-dependent, these components are model-level components.

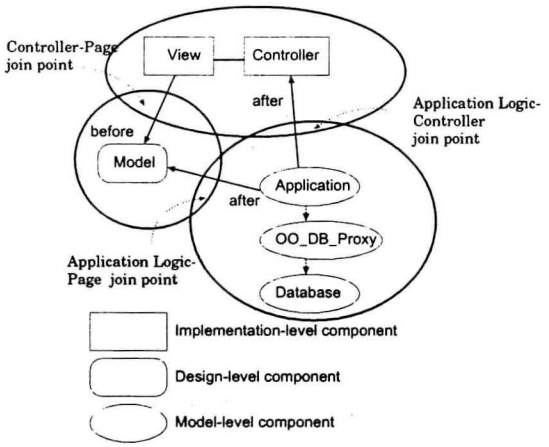


Fig. 5: Concrete View

3.4 Process View of WIS Architecture

Fig.6 shows a implementation process which can be drawn from the concrete view.

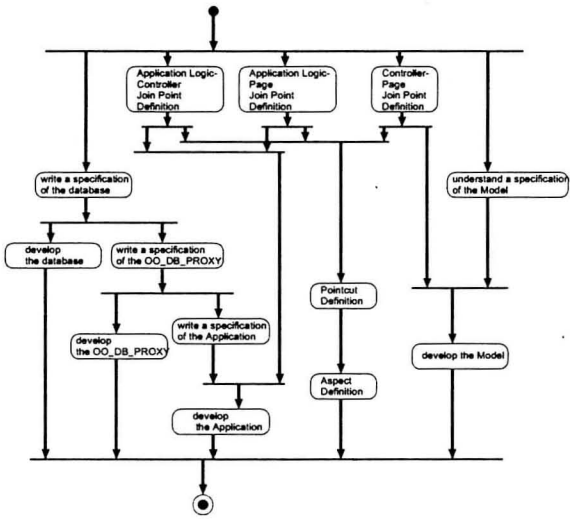


Fig. 6: Process View

4 Conclusion

We constructed an aspect-oriented software architecture for WIS. This software architecture implies concurrent implementation process for WIS, it can help us to select and coordinate Web-based technologies. As a result, it eases the development of WIS. Actually, we have developed WISs, and recognized this software architecture is useful for developing WIS.

UML Based Web Services Application Development Method

Jintae Kim, Eunjeong Kim, Mansoo Seo, Sooyong Park
Sogang University, REP. of KOREA
{canon, ejkim, msseo}@selab.sogang.ac.kr sypark@ccs.sogang.ac.kr

1. INTORDUCTION

The computing environments have turned from client-server environments to web environments. On the web, you can have images and information. But, these images and information cannot satisfy user's needs for more dynamic services. This causes new computing environments to emerge as Web Services. Web Services are service based applications on the web. This paper shows you relations between traditional software development method and Web Services by UML. Additionally, when you represent your Web Services application by DAML-S, this paper shows the transform rules between traditional software development method and Web Services based software development method.

2. RELATED WORK

2.1 Semantic Web

The Semantic Web is a vision: the idea of having data on the web defined and linked in a way that it can be used by machines, not just for display purposes but for automation, integration and reuse of data across various applications.

2.2 DAML-S

DAML-S (DARPA Agent Markup Language for Web Services) is service description language based on DAML which is resource description language on the Semantic Web. It has been developed for Web Services description on the Semantic Web. Our structuring of the ontology of services is motivated by the need to provide three essential types of knowledge about ServiceProfile, ServiceModel, and ServiceGrounding.

3. UML BASED WEB SERVICES DEVELOPMENT METHOD

3.1 The whole process

Usecase diagrams and specifications for requirements are made as well as Actor and usecase. The elements related with service profile are listed from usecase specifications by formalized transforming rules that are identified. The instance of meta- model makes class diagrams for ServiceProfile accomplish Figure 1 shows the whole process for UML based Web Services development method.

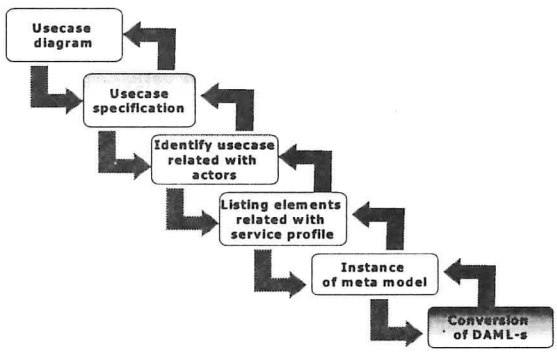


Figure1. The whole process for UML based Web Services development method.

3.2 Usecase modeling

Usecase modeling is user based and function oriented analysis method. That satisfies requirements of analysis method for Web Services. After usecase modeling, usecase diagrams and specifications are made.

3.3 Identification of Usecases related with actors

Actor is an external entity. Among usecases entirely choosen in system, usecases related with actors are identified and are used on list services' inputs, preconditions, outputs, and effects. Actors with this phrase lists of usecases are made.

3.4 List of elements in ServiceProfile

ServiceProfile is composed of two kinds of elements. Ones are to be identified by outputs through Web Services development process, the others are not because they can be identified not by development

process like analysis and design but by developer's experience. There are rules that identify inputs, outputs, preconditions, and effects in ServiceProfile.

- **Input, precondition**

① Identify usecases related with actors

② Identify inputs and preconditions from usecases' specifications, if actor sends messages or information to usecase

③ Iterate ② and make lists

④ Lists can be all of inputs and precondition in ServiceProfile

- **Output, effect**

① Identify usecases related with actors

② Identify outputs and effects from usecases' specifications, if usecase sends messages or information to actor

③ Iterate ② and make lists

④ Lists can be all of outputs and effects in Service Profile

UC_i: *ith* UseCase ($1 \le i \le n$, n is the number of UseCase in Service)
U : aggregation of candidate UseCases in Service

$input_i$: input of UC_i , $Output_i$: output of UC_i,
$precondition_i$: precondition of UC_i, $effect_i$: effect of UC_i

$input_{i\ (1 \le i \le n)} \in$ ServiceInput (ServiceInput : aggregation of $input_i$)
$output_{i\ (1 \le i \le n)} \in$ ServiceOutput (ServiceOutput : aggregation of $Output_i$)
$precontion_{i\ (1 \le i \le n)} \in$ ServicePrecondition (ServicePrecondition : aggregation of $precondition_i$)
$effect_{i\ (1 \le i \le n)} \in$ ServiceEffect (ServiceEffect : aggregation of $effect_i$)

$\rightarrow$: information of messages flow

If UC_i is related with Actors, $UC_i \in U$

($input$, $precondition$)
If Actor $\rightarrow UC_i$. $input_i \in$ ServiceInput and $precondition_i \in$ ServicePrecondition

($Output$, $effect$)
If Actor $\leftarrow UC_i$. $output_i \in$ ServiceOutput and $effect_i \in$ ServiceEffect

Figure 2. The formalized rules for listing the elements in ServiceProfile

3.5 Instantiation for ServiceProfile meta-model

3.5.1 ServiceProfile meta-model

Class diagram is proposed as meta-model for ServiceProfile as the following reasons.

First, class diagram is suitable for representing static properties of ServiceProfile.

Second, the structure of ServiceProfile *is similar to the structure of class.*

Actor class represents service provider or requester. It is independent class, so the information of service provider and requester is described as attributes.

Input, Output, Precondition, Effect and DomainResource classes are parts of functional description because there are several inputs, outputs, preconditions and effects in a service, which are represented as independent class.

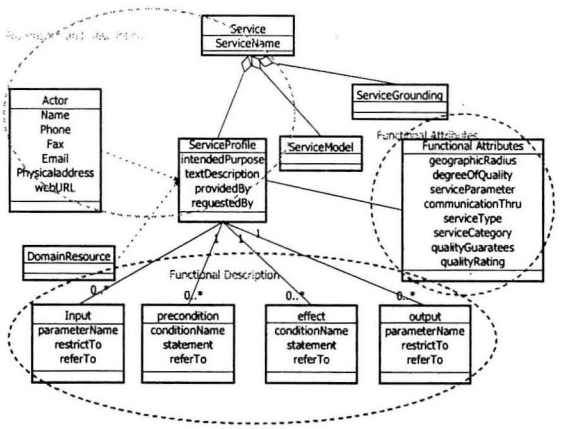


Figure 3. ServiceProfile meta-model

Functional attributes are also represented as one class because each element of functional attributes are simple so they can be correspond to one attribute of class respectively.

3.5.2 Instantiation for ServiceProfile meta- model

The list of elements in ServiceProfile is made by the rule that is proposed. ServiceProfile meta-model is instantiated by corresponding elements to the attributes of ServiceProfile meta-model. The instantiated ServiceProfile meta-model is converted to DAML-S.

4. CASE STUDY

Our approach will be explained through case study that is online book shopping service (Congo Service). Congo Service supports on searching and buying books on the web. Scenario of Congo service is as following: user input book information and user information (credit card number, address and so on). Using the information, user can get book information or buy the book.

5. CONCLUSION

UML based web services application development method was proposed in this paper. We proposed meta-model and rules for modeling DAML-S by class diagram in UML. It is worthwhile since it provided systematical development method for Web Services application on the Semantic Web. In the current, ServiceModel and ServiceGrounding are being studied and it will give many advantages to both developers and user on the Web Services environment.

Event Information Inputting and Publishing System developed by Social Science Students using XML & XSLT

Akira INOUE
Education and Research Center for
Information Science, Konan University
8-9-1 Okamoto,Higashinada-ku,Kobe,JAPAN
inoue-a@center.konan-u.ac.jp

Jun-ichi IGARI
Cyber Publishing Division
The Mainichi Newspapers
1-1-1 Hitotsubashi,Chiyoda-ku,Tokyo,JAPAN
igari@mainichi.co.jp

Ayako KOYAMA
The Graduate School of Policy and
Management,Doshisha University
Karasuma-Higashi-iru,Imadegawa-dori,
Kamigyo-ku,Kyoto,JAPAN
dbb0134@mail3.doshisha.ac.jp

Shigeo KANEDA
The Graduate School of Policy and
Management,Doshisha University
Karasuma-Higashi-iru,Imadegawa-dori,
Kamigyo-ku,Kyoto,JAPAN
skaneda@mail.doshisha.ac.jp

ABSTRACT

This paper demonstrates a practical *event information inputting and publishing system* for The Kyoto Branch Office of The Mainichi Newspapers, a major newspaper publisher in Japan. Graduate school students in social science develop the system. Real system implementation gives the students a good opportunity to study the impact of an IT system.The students, however, have poor ability in programming and little knowledge of computer science. To resolve this problem, we employ two tools in this project: XML (eXtensible Markup Language) and BXS (Baykit XML server), Japanese open-source software. The system is coded using XSLT–typed language "Xi (Extend it!^{TM})" included in BXS. The total amount of XSLT coding steps is less than 3,000 lines. This means that XML is not a "high fence" BPR tool, but is a "simple use tool for programmers" that simplifies the software system development process and decreases the total amount of software coding.

Event Information Inputting and Publishing System

Kyoto is one of the most famous sightseeing spots in Japan. Many *events* (exhibitions, festivals, shrinesopening to public, and so on) are held there. Information on these events seems to be not only of worth for those who live in Kyoto, but also useful for tourists. We investigated event information as contents sent via the Internet. Graduate school students in social science developed event information inputting and publishing system.

Figure 1 is the top page of Mainichi Newspapers' Kyoto branch office homepage, "Kyoto Everyday." A graduate student designed this top page. "Kyoto Everyday" has two types of event information page: an "Event List" table shown in Fig. 2, and "Event Detail" tables, one of which is shown in Fig. 3.

The homepages of Figures 2&3 can be automatically generated by the event information inputting and publishing system that was developed by the students. This system is a kind of Web-application system. Figure 4 shows the block diagram of the developed system.

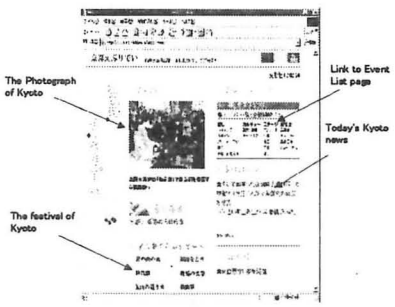


Figure 1: "Kyoto Everyday" Toppage

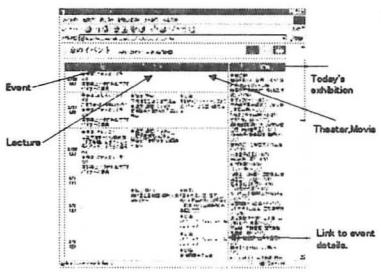


Figure 2: "Event List" Page

Application of XML & BXS (Baykit XML Server)

Social science students have less skill in computer programming and Internet technologies. XML (eXtensible Markup Language) and XSLT (eXtensible Stylesheet Language Transformations), however, made it possible for the students to implement a real Web application system in a short period of

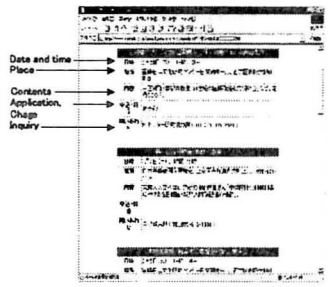


Figure 3: "Event Details" Page

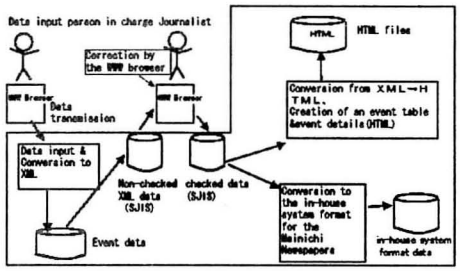


Figure 4: Event Information Inputting and Publishing System

time.

Another easy implementation tool is "BXS," (Baykit XML server), an open-source software developed in Japan. BXS has a specially designed language "Xi," (eXtend It) to convert XML data to another format. Xi is an extension of XSLT and the proposed system was coded using Xi. Coding by Xi accounts for about 3 Kline(s) in the whole system.

BXS BXS is an open-source software developed by Yokohama Baykit, a Japanese open-source community. BXS has an XML processing Server and an HTTP Server. According to Yokohama Baykit, the features of this software are as follows.
(1) BXS has XML processing modules and Web Server modules. It can easily access XML documents.
(2) BXS is Japanese open-source software. It has Japanese documents, mailing lists, and BBS. It is can easily deal with Japanese characters.
(3) BXS installation is easy. It can run on Windows, Linux, and MacOS.

Xi Xi is an XML document generation language that Yokohama Baykit has developed. There is an Xi interpreter engine which interprets Xi and generates an XML document. In Xi, an XML format describes the logic for generating XML like XSLT.

[Xi sample program]

```
<xi:program
xmlns:xi="http://www.baykit.org/Xi/1.0">
<HTML><BODY>
Zipcode is
<B>
<xi:value-of
select="$Web.parameters.zipcode"/>
</B>
.
<BR/>
<xi:variable name="now">
<xi:value-of
select="$Java.new('java.util.Date')"/>
</xi:variable>
Present time is <B>
<xi:value-of select="now.toString()"/>
</B> o'clock.
</BODY>
</HTML>
</xi:program>
```

Essential Leaders for University-Industry Cooperation
Almost all the members of this project are graduate students who are not specializing in computer engineering. In spite of this disadvantage, this project went well. It is true that such a useful tool as XML or BXS must have been one of the important elements in our success, but the existence of project leaders was also essential. There were two leaders in our team. Each leader had the following role and talent.
Top programmer: a person who is a fully-skilled programmer with extensive experience of system engineering and programming.
Producer: a person who is eager, as a coordinator to make the project a success, besides explaining the business process from the customer's perspective.

Conclusion
This paper has demonstrated a new approach: real application system development by graduate school students in social science. Students developed a web application system for event information inputting and publishing for Mainichi Newspapers, a major newspaper publisher in Japan.

The major tools used for easy implementation were XML and BXS (Baykit XML server). XML was not difficult to input, because XML is just text data. Programming language Xi of BXS, an extensional XSLT, was a powerful tool to convert XML data to another format. This means that XML is not a "high fence" BPR tool. XML is a "simple use tool for programmers" that simplifies the software system development process and decreases the total amount of software coding.

Paper title : MDR based Network Object Management System

Authors : O-Hoon Choi*, Doo-Kwon Baik.

Address : 225, Asan Science Building, KOREA UNIV., 1, 5-Ka, Anam Dong, SungBuk Gu, KOREA Univ., Seoul, Korea.

Affiliation : KOREA UNIV.

e-mail : pens@software.korea.ac.kr

Phone : 82-2-925-3706

Fax : 82-2-921-9137

Topic area of paper : Information Management Systems and Management Information Systems Track

ABSTRACT

In order to satisfy the complex and various demands of the customers, telecommunication networks must provide their services rapidly and flexibly, with the support of efficient service and network management systems. To satisfy these requirements, many managed objects have appeared to manage network services and elements. But there is no way to manage these new managed objects efficiently in object-oriented software. Therefore we study several problems when subsystem using object-oriented technique is implemented. First problem is that interface works between realized application programs using its technique. Second problem is that reusability of internal managed objects are difficult.

In this paper, we will point out some problems. And to solve the problems, we have proposed the Object Management System (OMS) architecture, which supports transparent interface between object-oriented applications and the distributed data repositories. Also in order to manage efficiently business objects stored in the different repositories, we have suggested a method for integration of dynamic information resources in heterogeneous and

the distributed network environments. Finally, we applied OMS to TINA domain, implemented OMS, verified it with SMV.

. Object management System

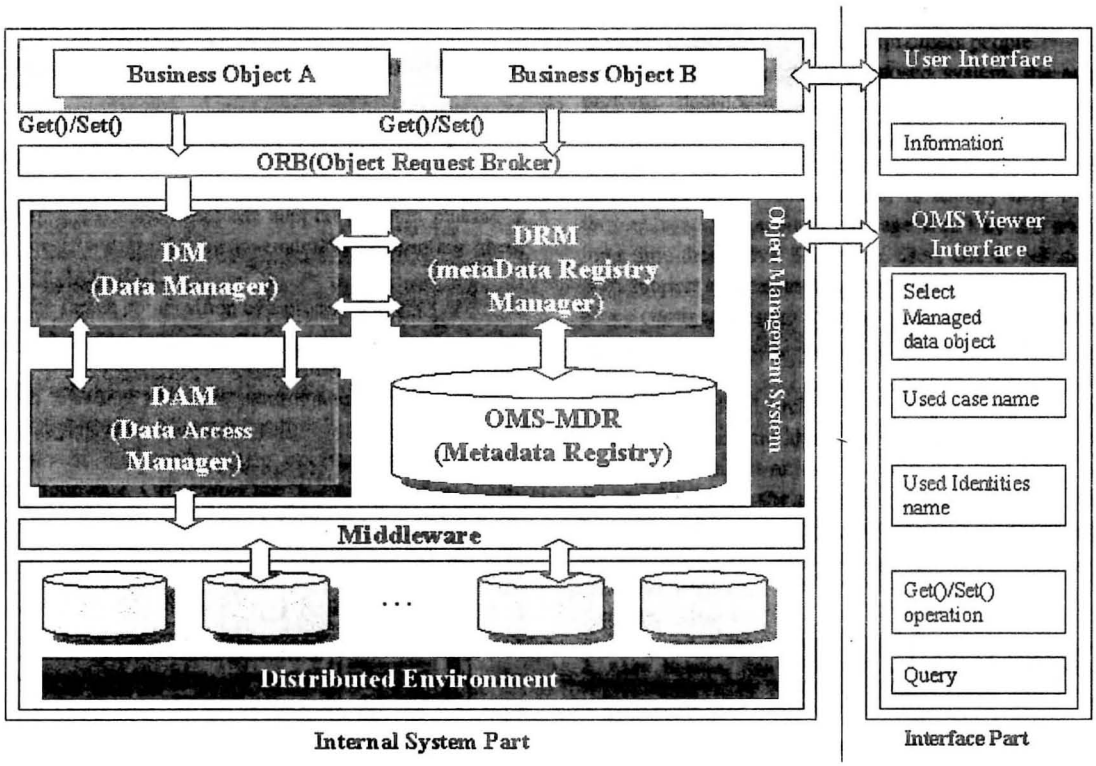
In this chapter, we explain the architecture of OMS proposed in this paper and the method of modeling. And we explain the main idea of Object Management Operator(OMO), which is the main process of OMS. We designed the proposed system with UML and implemented it based on CORBA.

The purpose of OMS is a separation of object oriented applications and middleware. It also stores and manages the business objects in the metadata registry. The OMS framework is devised to ensure data type and location transparencies without replying on code generation technique. It is our intent to provide data

access transparency in OMS. In some sense, the OMS can be regarded as a superset of data interface middleware. In this paradigm, the role of OMS is to convert data retrieved by data interface middleware into managed objects. Application Layer and OMS communicate using the object middleware CORBA and they provide an interface language IDL(Interface Definition Language) that is defined regardless of client language, server language, and platform.

Design of OMS Architecture

OMS consists of three layers. Theses layers are Object Management Operator Layer, Business Object Layer and Distributed Environment Layer. The Business Object Layer consists of a set of Business Object and a interface for the Object Management Operator Layer. The Object Management Operator Layer is a set of processes for processing requirements through CORBA based interface. The Distributed Environment Layer consists of a real data in RDB. Figure 1 shows the architecture of OMS.



<Figure 1> Architecture of OMS

Atomic Commitment Protocol for Distributed Real-time Transactions

(SUMMARY)

YunSheng Liu **GuoQiong Liao**

(Computer Institute of Huazhong University of Science & Technology, Wuhan 430074, P.R.China)

ysliu@hust.edu.cn joanliao@263.net

1. Introduction and Related Works

For the rapid progress of network technologies, applications of real-time database systems (RTDBS) are blossoming in distributed application environment. But very few works have been done on the commitment of distributed real-time transactions until now. The most difficult about it is the transactions in real-time database systems must satisfy timing constraints besides ACID properties. The atomicity of real-time transactions is more difficult to be guaranteed than general distributed transactions.

Traditional atomic commit protocols, such as Basic Two-Phase Commit Protocol, Optimistic Two-Phase Commit Protocol, Three-Phase Commit Protocol, etc, are not real-time commit protocols and can't guarantee the atomicity of distributed real-time transactions. Several real-time commit protocols, such as PROMPT, PEP and RCP, are also can't guarantee transactions' atomicity at any time. Therefore, it is necessary to develop new commitment protocol to ensure the atomicity of distributed real-time transactions.

In this paper, pour works can be concluded as follows:

1) The correct criteria for the atomic commitment of distributed real-time transactions are given;
2) The real-time atomic commit protocol (RTACP) is discussed in details;
3) The performances of RTACP are analyzed and compared with other real-time commit protocol.

2. Atomic Commit Criteria for Distributed Real-Time Transactions

Let ST be the set of participating sub-transactions of T, i.e., ST = $\{t_i \mid t_i$ is a sub-transaction of T, which is executing at the participating site$\}$; CT be the set of committed sub-transactions of T, i.e., CT = $\{t_i \mid t_i \in ST \wedge t_i$ has committed$\}$.

Definition 1. *A commit protocol is said to be an atomic commit protocol (ACP), if for any transaction T it can guarantee either* $\forall t_i \in ST(t_i \in CT)$ *or* $\forall t_i \in ST(t_i \notin CT)$

In general, if a commit protocol is an *atomic commit protocol*, it must satisfy following criteria:

ACC1. *All participants of a distributed transaction that reach a decision reach the same decision.*

ACC2. *The sufficient and necessary conditions of commitment for all the participants are that all of them have reported REQUEST FOR COMMIT.*

ACC3. *Each participant reports at most once, i.e., the report can not be changed.*

ACC4. *The coordinator makes decision at most once, i.e., the decision can not be changed.*

However, above criteria are not sufficient for distributed real-time transactions to ensure atomicity since participants may change their decisions by missing deadlines. Therefore, two real-time criteria should be added:

RTACC1. *If a participant has not received any decision from the coordinator after reporting REQUEST FOR COMMIT, then it can not unilaterally commit or abort even if it has expired.*

RTACC2. *If the coordinator has made a COMMIT decision, then the distributed transaction is considered to be committed.*

3. Real-Time Atomic Commitment Protocol (RTACP)

Before introducing RTACP, two definitions are given as follows:

Definition 2 *The deadline of the coordinator receiving report from the participants is* $d_A = d_T + \delta$, *where* d_T *is the deadline of the distributed transaction T and* δ *is the normal maximum communication delay of network.*

Definition 3. *For any sub-transaction (including coordinator)* t_i, *the time when it becomes successful is called its "committable point". After* t_i *reaches the "committable point", it is called that* t_i *has been in "committable state".*

RTACP can be described as following rules:

1) If any participant comes into committable state, then it reports REQUEST FOR COMMIT to the coordinator independently. Since then the participant must wait the coordinator' decision in spite of the timing constraint.
2) If any participant comes into uncommittable state, then it reports ABORT to the coordinator and aborts actively.
3) Once the coordinator has received REQUEST FOR COMMIT reports from all participants before d_A, it makes COMMIT decision and sends the decision to all participants. After sending the decision, it is considered as the distributed real-time transaction has committed. (RTACC2)
4) If the coordinator has received at least one ABORT report before d_A or has not received all REQUEST FOR COMMIT reports until d_A, then it decides ABORT and sends this decision to the participants which have not reported ABORT (not to all the participants). After sending ABORT decision, it is considered as the distributed real-time transaction has aborted.
5) Once receiving a decision from the coordinator, the participants must perform the decision in spite of timing constraints.

4. Performance Analysis OF RTACP

The results of performance comparison of several real-time commitment protocols are shown as following table:

indexes / decision / protocols	The number of logging record		The number of transferred message		atomicity guarantee	transaction semantic
	commit	abort	commit	abort	—	—
PROMPT	4N	4N	2N+1	2N+1	non	firm deadline
RCP	2N	N+Nc	2N+1	N+Nc+1	non	compensatable
RTACP	2N	N+Nc	2N+1	N+Nc+1	yes	firm and soft deadline

5. Conclusions

Motivated by the need to guarantee the atomicity of distributed real-time transactions and reduce the overheads of commit processing, a new real-time commit protocol called RTACP for firm or soft deadline transactions is proposed in the paper.

RTACP is different from the general commit protocols:

1) RTACP requires the commitment of real-time transactions must obey the real-time atomic commit criteria;
2) RTACP eliminates the phase of initiating vote in those protocols.
3) RTACP endows the participants and the coordinator the ability to abort independently.

RTACP is also different from other real-time protocols.

1) The most important advantage of RTACP is it can guarantee the atomicity of real-time transactions.
2) Moreover, the system performances can also be improved by RTACP.

Analyst-type Secretary Agent System for Schedule Management -Toward Sociality Sensitive Communication Support-

Hiroshi Igaki[†] **Makoto Sakai**[‡] **Hajimu Iida**[†] **Ken'ichi Matsumoto**[†]

[†]Nara Institute of Science and Technology
8916-5, Takayama-cho, Ikoma-shi, 630-0101 Japan
{hiro-iga@is, iida@itc, matumoto@is}.aist-nara.ac.jp

[‡]SRA Key Technology Laboratory, Inc.
3-12, Yotsuya, Shinjuku-ku, Tokyo, 160-0004, Japan
sakai@sra.co.jp

1. UBIQUITOUS-COMPUTING SOCIETY

Recently, it becomes universal for individuals to possess two or more internet accessible information terminals. So far, the ubiquitous-computing society [3] is being realized. People belong to various communities according to their work or private activities. Schedule management work, which determines the matter "when to meet", "whom to meet", and "for what purpose to meet" among two or more persons, considering the various senses of values to a user's society, is typically carried out using communication tools in the cyber world. On the other hand, communication delay of the system such as E-mail was so conventional problem. So the message agent systems, which automatically and intelligently responds to received message instead of a user, has been studied [1][2][5][13][14] in order to resolve this communication delay problem.

2. RELATED RESEARCH

For example, As for "Saeko" [1], and the system in [14], schedule management agent generally holds all users' schedule data. Based on this data, the agent system determines meeting plan using specific algorithms. Such a system can perform efficient schedule management for a group working closely together. However, the schedule data of all persons concerned must be inputted into groupware as a premise. The answer to the invitation is determined only from existing free time of personal schedule data.

The Meeting Scheduling Agent [5] perform automatic answering based on machine learning(Fig.1).Then, the agent can perform auto-answer to a request sender based on the study result.

However, this technique essentially has the following restrictions.

In order to make an auto-answer, a user needs to repeat to answer manually in order to let the agent learn beforehand.

Contents of auto-answers are not always guaranteed as desired.

In various communications in everyday life, some incorrect decision may have the very expensive cost. So, we call the automation-oriented agent system which performs schedule management instead of a user "Deputy-type secretary agent system." We propose a new concept of "Analyst-type secretary agent system".

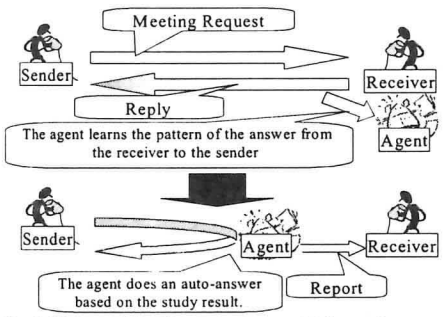


Fig.1 The model of the system, which performs an auto-answer based on the machine learning

3. ANALYST-TYPE AGENT SYSTEM

"Analyst-type agent system" collects information related to decision making, arranges, and presents to the user as supporting information. This proposed system is based on the following policies:

a) The user should make most decisions. His/her agent does not perform the auto-answer which may not accompany user's mind.

b) The agent just may make a decision with a very low possibility of being contrary to the intention of users, as fixed processing.

c) The agent collects and arranges data supporting decisions that will be made by the user.

To realize these policies, on proposed system, at first, we formally define the schedule management work treated in this paper as follows[4][7].

There are the following two kinds of meeting requests, and therefore, there are two kinds of rejections:

Participation request: The meeting plan (such as the start time, etc.) is announced by the client (requester of the meeting), and the user just reply his/her attendance or absence.

Prior inquiry: Selectable list of the meeting plans is shown, and the user reply his/her attendance with selected meeting plan, or the user reply his/her absence.

The workflow of "Prior inquiry" reception was shown in Fig.2 using StateChart. In this workflow, each state transition ignites by

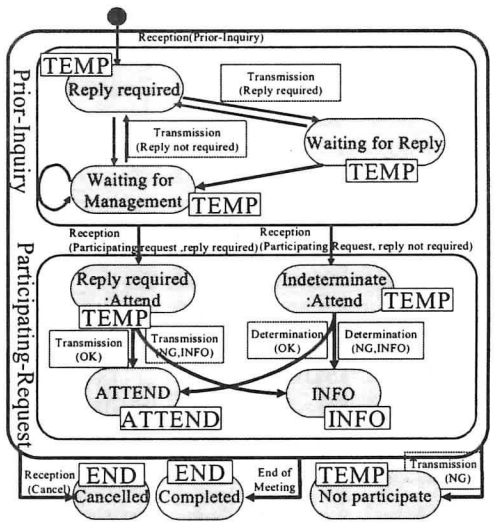


Fig.2 The schedule management workflow at the time of "prior-inquiry"

the event corresponding to the user's operation.

By looking over the progress status arranged for all meeting request currently managed, the user realizes the current condition of his/her agenda. The information prepared as a context is categorized to the following two kinds.

[A-1] The status of each meeting: An agent decides the current state in the workflow of the received meeting request.

[A-2] The status summary of the whole schedule: The agent lists the state of all meeting requests. The agent records the data (exchanged messages) of the past meeting requests, and arranges this data relevant to the present request based on three items: 1) purpose of the meeting, 2) client (originator) of the meeting, and 3) owner group of the meeting. Then the agent shows arranged data as history information. Specifically, the following information is shown:

[B-1] The number of *Related histories:*Related history means an item (meeting request) in the past,

[B-2] Attended times and absence times: Attended times and absence times counted from Related histories

[B-3] Actual arrangements in *Related histories:*Actually made arrangements (e.g. change of date & time) found in ***Related histories***.

By looking the contexts of ***related history*** (when it was carried out, and how it was managed), the user gets to be able to understand following situations easily:

1) How many times did I attend the past meeting of the same kind?

2) Whose intention cancelled similar meetings in the past?

3) When were the meetings of the same kind held?

4) Can any correspondence send in the past be reused this time?

An Analyst-Type secretary agent can support decision making at the time of user's schedule management by presenting these information ([A-1..2], [B-1..3]).

The Fig.3 is the screen composition of a proposed agent system. In the architecture of the proposed system, each user has his/her secretary agent. The agent supports communication between users using a WEB browser. The agent coordinates between multiple users by exchanging semi-structured mail.

The main part of the agent system operates as add-in of a commercial groupware OfficeWorkTop [6] developed at Sharp Incorporated Company.

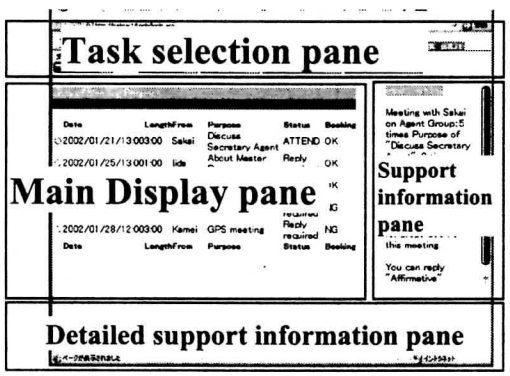


Fig.3 The screen composition of a proposed system

4. Example of Decision-Making Support on Receiving of "prior-inquiry"

By looking every **Status** which are presented by the agent, user can realize the current status of each message and currently required works (i.e. required decisions).

User chooses the message by Status "Reply Required" (Fig.3). Furthermore, the link to the current schedule information and the related meeting held in the past are displayed on the support information pane. Thereby, a user makes decisions to the received meeting request by reviewing support information.

For example a user can make the following decisions from the history information extracted from the related history.

Since the user cancelled the request from ***Client*** A for the sake of a user's convenience before, he/she should attend this time.

Although the user always refuses the meeting request from ***Client*** B, he/she should attend this time since this request is sent to whole members of the affiliation of the user.

The agent supports a user's decision-making as mentioned above. Based on the information shown by the agent, the user can make decisions easily in variegated situations.

5. DISCUSSION

If the conventional system's agent made mistakes in the answer, the user has to correct it. At this time, the user may have loss of social trust in some situation. It is because schedule management is social cooperative working with others people.

On the other hand, in our proposed system, the agent does not perform decision-making on the matter related to social responsibility. Furthermore, the wide-use of PCs and PDAs may reduce the demand for the full-automatic processing by the agent in contrast. It is because a user is usually near by information terminals, so the received message won't be left unread so much long time these days. Therefore, users check messages frequently with support of Analyst-Type agent system to make right decision.

Since our agent system is based on an actual secretary's policy and its reference model, this kind of support will be made so efficiently[4][7].

Network-based schedulers are widely used now. However, these systems seldom care about the sociality or security; they are just for closed members or just for public use. One of the key factors to be considered for the schedule management over the Internet is consideration of sociality. Our proposed system will have advantage for daily and casual use for urban ubiquitous computing environment.

REFERENCES

1. A Secretary Agent "Saeko," http://www.scientia.co.jp/products/sae.html
2. Akira Kataoka, Koji Kida, Hiroyuki Tarumi, "Meeting Scheduling With Negotiating Agents," No.54, Japan conference of IPSJ, 1997.
3. Mark Weiser, "The computer for the twenty-first century," Scientific American, pp.94-104, September 1991.
4. Makiko Nishizawa, "Executive Secretary," Hakuto Shobou, 1991.
5. Pattie Maes, "Agent that Reduce Work and Information Overload, " Communications of the ACM Vol.37, No.7, pp.31-40, July 1994.
6. Sharp Incorporated Company, "Web Gropware Office WorkTop," http://www.sharp.co.jp/owt/
7. Zenkokudaigaku Tankidaigaku–jitumukyouikukyoukai, "Secretary Task," Kinokuniya-Shoten, 1998.

(Other references are written in Proceeding)

A Simplified Graph Model for User Interface Constraints

Chuan-chieh Jung
National Central University, Taiwan
jcc@db.csie.ncu.edu.tw

Tze-Heng Ma and *Y. S. Kuo*
Academia Sinica, Taiwan
{mada | yskuo}@iis.sinica.edu.tw

Constraints have been used extensively for the construction of graphical user interfaces. For instance, a graphical user interface may contain a line segment that connects two movable points. When a user moves one of the points, the line segment is required to move along with the moving point. In this case, constraints exist between the end points of the line segment and the movable points. In general, constraints are relations on user interface objects specified by interface developers, which are automatically maintained by the system. A graphical user interface may contain hundreds or even thousands of inter-dependent constraints. A constraint system frees interface developers from the drudgery of maintaining these relations continually.

Constraints used in graphical user interfaces typically have a dataflow-like network model [4] independent of the functionality of constraints. Constraints are relations (e.g. equations) on variables. A constraint is associated with a small set of methods; each of the methods can serve to satisfy the constraint. For instance, by treating all its variables except one as constants, an equation typically has an associated symbolic solution, which is a method. Constraint planning refers to the selection of appropriate methods to satisfy constraints. Since methods may conflict with each other, constraint planning algorithms are aimed at avoiding the conflict.

Investigations into constraint planning algorithms have been toward not only efficiency but also generality. Early algorithms are restricted to constraint systems where the constraint graph is a tree and all methods have a single output [1,2,5]. As they have evolved toward generality with these restrictions removed, constraint planning algorithms have become highly complicated. (The recent work of QuickPlan [6] is 43 pages thick.) This may attribute to the inherent complexity of the constraint model, which involves three entities: variables, constraints and methods. To overcome the complexity, we propose a simplified graph formulation of the problem. The new problem formulation is as general as the conventional one [4] but is much simpler for it involves just one entity: the methods. We will show that the constraint planning problem can be reduced to a well-known graph-theoretic problem, finding feedback

vertex sets, on the simplified graph model.

Based on the simplified problem formulation, we are able to develop simple algorithms for constraint planning under a most general setting. The algorithms allow methods that have multiple outputs and can be applied to constraint graphs that contain cycles. The algorithms even allow constraints whose methods do not reference all variables of the constraints. (Such a constraint is referred to as non-uniform in contrast to an ordinary uniform constraint.) This last point makes the algorithms more general than all previously proposed algorithms. The authors admit that non-uniform constraints are infrequent in practical applications. However, they are not unrealistic as illustrated by a simple example.

Constraint systems are referred to as one-way or multi-way depending on whether each constraint is associated with just one method or more [5]. Multi-way constraints are more general and declarative than one-way constraints. The process for satisfying multi-way constraints typically consists of two phases. In the first phase, an evaluation plan is constructed that selects methods for constraints. The second phase then executes the selected methods with a one-way constraint satisfaction algorithm [3] to actually satisfy the constraints. This article only considers the multi-way constraint planning phase.

References

1. B. N. Freeman-Benson, J. Maloney, and A. Borning, An incremental constraint solver, Commun. ACM, 33, 1(Jan. 1990), 54-63.
2. M. Gangnet and B. Rosenberg, Constraint programming and graph algorithms, Annals of Math. and Artificial Intelligence, 8, 1993, 271-284.
3. S. E. Hudson, Incremental attribute evaluation: A flexible algorithm for lazy update, ACM Trans. Program. Lang. Syst., 13, 3 (July 1991), 315-341.
4. J. H. Malony, Using constraints for user interface construction, Ph.D. Thesis, Dept. of Computer Science and Engineering, Univ. of Washington, Seattle, Wash., 1991.
5. M. Sannella, J. Maloney, B. Freeman-Benson and A. Borning, Multi-way versus one-way constraints in user interfaces: Experience with the Delta Blue algorithm, Software Practice and Experience, 23, 5 (May 1993), 529-566.
6. B. Vander Zanden, An incremental algorithm for satisfying hierarchies of multi-way dataflow constraints, ACM Trans. Program. Lang. Syst., 18,1(Jan. 1996), 30-72.

Paper title : Metadata Based Virtual Schema Supporting Decision Making

Authors : JuHum Kwon*, Doo-Kwon Baik.

Address : 225, Asan Science Building, KOREA UNIV., 1, 5-Ka, Anam Dong, SungBuk Gu, KOREA Univ., Seoul, Korea.

Affiliation : KOREA UNIV.

e-mail : jkweon@software.korea.ac.kr

Phone : 82-2-925-3706

Fax : 82-2-921-9137

Topic area of paper : Data engineering and Information Management Systems

Abstract

Several fundamental approaches have been proposed for database integration such as database federation, global schema, mediator architecture, information brokering, and data warehouse. The main design goal of those systems focuses on the information integration by combining semantically associated objects. Those schema integrations, however, do not completely address the two important issues. One is the global operation such as union, intersection, and difference and the other is semantic conflict. Some of them partly address the issues but not complete. That is because it is their limitation to address all issues together.

In this paper, metadata based virtual schema is proposed to integrate heterogeneous databases, where global operations can be computed as precise as in the single schema from the virtual schema and also solves the semantic conflicts using metadata representation.

Metadata based Virtual Schema Architecture

As shown in the figure 1, the proposed system consists of three major components: Virtual schema, Metadata representation, and Query manager.

• Virtual schema

Virtual schema is a schema that a set of descriptions of various local databases modeled with an ERD. The descriptions are stored in the metadata representation containing information for every participating local object, object relationship, and combination rule. The virtual Table (VT) can be defined as follow:

$$VT(\bar{X}) = \bigcup_{1 \le i \ge n} R_i(\bar{Z}_i), C_Q$$

Where:

1. R_i is a relation in the local schema.

2. Z_i is any set of participating attributes in the R_i
3. C_Q is a conjunction of order subgoals of the form $u\theta v$, where $\theta \in \{<, >, \leq, \geq$, and relational operators$\}$

and $u, v \in \bigcup_{1 \leq i \geq n} \overline{Z}_i$, $\overline{X} \subseteq \bigcup_{1 \leq i \geq n} \overline{Z}_i$

• Metadata representation

The descriptions of every local database object participating in virtual schema design are stored in the metadata representation. It contains, for example, local database object, object type, semantics of object, the description of virtual table and virtual attribute, object combination rule, and so on.

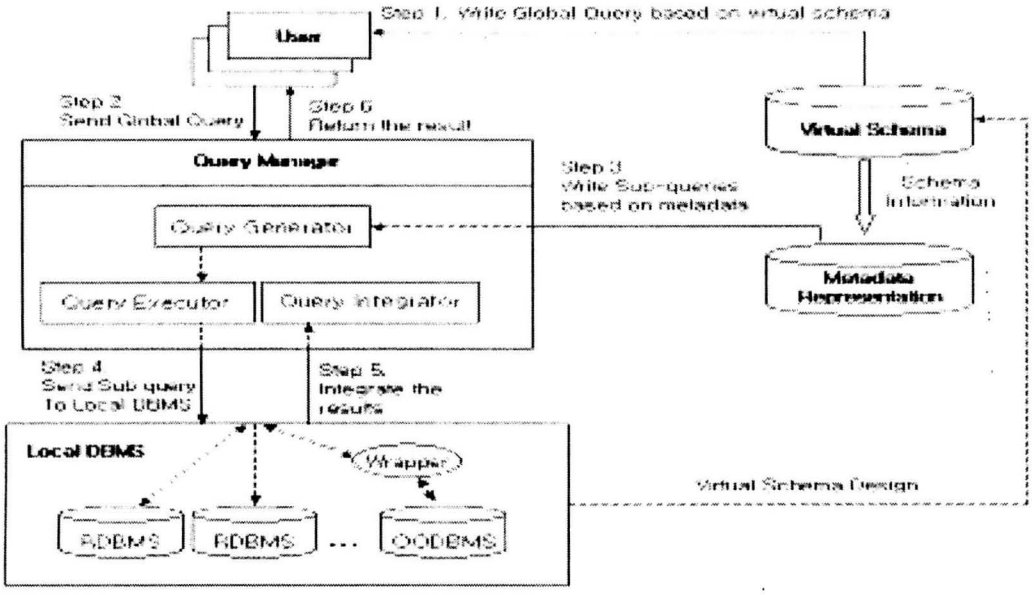


Figure 1. The principle of Metadata based Virtual Schema

• Query manager

The query manager is responsible for writing local source database specific sub-query from the given user global query and issues the sub-queries to the local databases. Its other job is to integrate local database query results.

A Decentralised Architecture for Workflow Support

Jun Yan, Yun Yang and Gitesh K. Raikundalia
CICEC-Centre for Internet Computing and E-Commerce
School of Information Technology
Swinburne University of Technology
PO Box 218, Hawthorn, Melbourne, Australia 3122
Email:{jyan,yun,gitesh}@it.swin.edu.au

SUMMARY

According to the Workflow Management Coalition (WfMC), workflow is the automation of a business. It schedules the various partially-ordered tasks of a business process to reach a goal. As an enable technology for Business Process Reengineering (BPR), workflow has attracted the attention of many researchers, vendors and users, and experienced tremendous growth in the last decade.

Theoretically, workflow has demonstrated great power in supporting business processes, but, in practice, many previous attempts at using workflow systems failed, as they exhibit common weaknesses, such as bad performance, poor scalability and user restrictions. From a system architecture perspective, most of these problems result from the mismatch of application requirements and conventional approaches. Given the nature of the application environment and the technology involved, workflow applications are inherently distributed. But, because of the benefits like easy auditing and design simplicity, most existing workflow tools, use centralised client/server software architectures. Although many research efforts such as Exotica/FMQM, Endeavors, DartFlow, etc. have addressed the architecture issues, however, all these approaches failed to eliminate the central data repository and central workflow engine from workflow system. Thus they only partly addressed the above difficulties and the problems are still unsolved.

This paper proposes an innovative, decentralised workflow architecture named as SwinDew, which applies peer-to-peer (p2p) concepts to workflow scenarios and decentralises both data and control. In this approach, a workflow system is organised as a p2p system with each node in the system being a self-managing, independent part from both data and control viewpoints. Compared with conventional workflow systems, each node plays the role of a client and performs part of the server functions. As shown in figure 1, each node

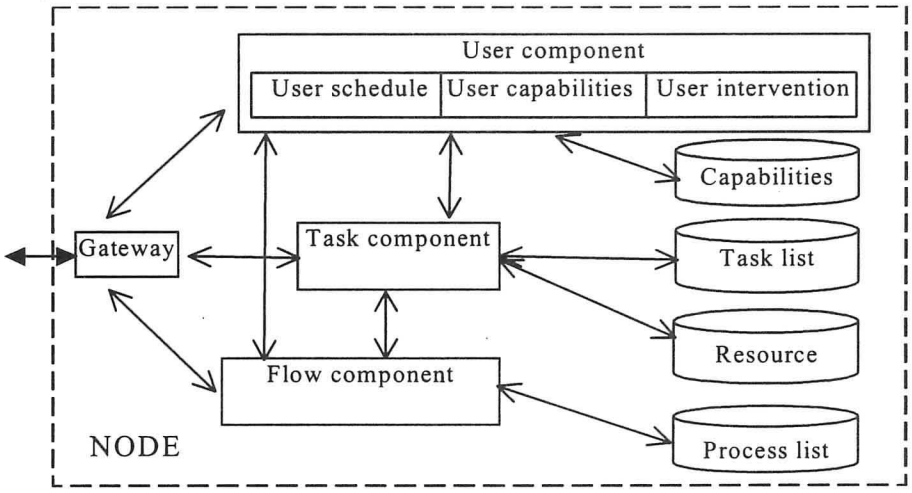


Figure 1: Node structure in a decentralised workflow system

consists of *user component, task component* and *flow component*. The *User component* provides a friendly and powerful user interface and deals with user-related functions. The *task component* residing in a specific node is in charge of the execution of instances created at that node. And, the *flow component* administers relationships among the tasks. These three components interact with one another to carry out node functions. At the same time, each node keeps a part of the system data, which is controlled by the three components, respectively. In addition, through a common gateway, one node can exchange information with other nodes, either one-to-one or one-to-many.

To support this decentralised workflow architecture, some mechanisms are offered to carry out build-time functions and run-time functions. In build-time, after a process is defined at a single node, the basic idea is to divide a process into individual parts and distribute them to appropriate nodes. To achieve this goal, a six-tuple *T* is used to describe every task and its relationships in a specific process:

$$T\ (processid;\ taskid;\ C_{pre};\ C_{post};\ capability;\ resourceset)$$

With the support of this description, individual tasks can be separated from a process easily. Based on the capability index, the tasks can be sent to appropriate nodes. Later, with the unique process identifier, it is very easy to unite various tasks at dispersed locations to form a complete process. Regarding run-time functions, the paper addresses the instance creation and instance execution. Process instances are created gradually with the creation of task instances and tasks are assigned on-the-fly through the communications between related nodes. Finally, the system does not rely on a single workflow engine to coordinate execution. Instead, process execution is controlled with the efforts of all the components from start to termination.

In summary, in workflow systems, performance bottlenecks, poor scalability and user restrictions are common weaknesses caused by using traditional client/server architectures. So far, these issues have not been addressed well. This research aims at solving the above problems thoroughly by removing central servers from the workflow system completely. In this paper, SwinDeW–~~a~~ decentralised architecture for workflow support–~~has~~ been proposed, which applies p2p concepts to workflow scenarios. In this approach, both data and control are distributed and process execution relies on the coordination of individual nodes. The corresponding process representation, build-time functions and run-time functions have also been described. With the support of this approach, workflow systems can be built in a truly distributed fashion without the help of a central data repository and a central workflow engine. Therefore, potential single points of failure are eliminated and node failures will not stop the whole system. The size of the system can scale from dozens to thousands without difficulty in a dynamic environment. And the users are better supported.

Support for Maintaining Distributed Component-based Systems

Xuefen Fang
SRA Key Technology Laboratory, Inc. &
Dept. of Graphics and Computer Science
Graduate School of Arts and Sciences
The University of Tokyo
fang@graco.c.u-tokyo.ac.jp

Tetsuo Tamai
Dept. of Graphics and Computer Science
Graduate School of Arts and Sciences
The University of Tokyo
tamai@graco.c.u-tokyo.ac.jp

1 Objectives and Motivation

This dissertation is concerned with the difficulties of understanding large, distributed component-based software systems, a system understanding is a major activity in maintenance process. Specifically, it is interested in reverse engineering techniques and tools that can help to recover the architecture of a existing distributed component-based system. Relatively recently, distributed component-based software systems are getting more and more in service, developed under a tight schedule, with a high employee turnover, and in a rapidly evolving environment, and subject to successive changes throughout their working lives, which diminish their understandability. Such a system is hard to maintain from the very beginning. Therefor, it is the purpose of this dissertation.

- to investigate and propose practical techniques that can aid maintainer in identifying the architecture of an existing distributed system; and
- to show (through detailed case studies and analysis of related work) that such techniques help to better maintenance practice.

One major reason for this concern is that distributed component-based systems present unique maintenance challenges. Traditional maintenance involves mostly observing and modifying lines of source code. An aid to program understanding, typically identifies a decomposition of the system into a hierarchy of subsystems and modules, and shows interactions between these subsystems and modules. However, In distributed component-base systems, the primary unit of construction is generally component, which are based on a component model and is limited visibility. Maintainers must deal with most of the system at the component level and the component implementation environment. For that purpose, understanding such system is concerned with issues pertaining the overall system organization: global control structure; protocols for communication and data access; assignment of functionality to design elements; physical distribution and so on. Those are described as system architecture.

2 Architecture Views

In fact, when maintainer want to understand a system, the first thing they will do is to reconstruct architecture views from that system, because of software architecture is most usefully mental model shared among the developer and maintainer. According to observation of practitioners, the following views are indeed effective in understanding distributed component-based systems.
Conceptual Architecture View: A conceptual architecture view consists of conceptual components linked together to deliver the functionality of the application. In a distributed component-based system, there are explicit conceptual views. This view is usually tied closely to the infrastructure model. Dependent on the conceptual view, The software basic architecture of a system can be specified.
Code Architecture View: A code architecture view describes how the software implementing the system is organized. In this view, individual elements are abstracted from source components and deployment components.
Execution Architecture View: An execution architecture view describes the control flow from the point of view of the runtime platform. This view helps the maintainer to understand the following concerns.

- How does the system meet its performance requirements?

- What are the impacts of a change in the runtime platform?

As systems are distributed, the developers need to understand how functional components map to runtime entities, and how communication, coordination, and synchronization are handled.

Component Architecture View: A component architecture view is used to make explicit how the functionality is mapped to the components, and what relationships are. Component architecture view can help the maintainers to understand the following concerns.

- What component is mapped to which server
- What dependencies between components exist

3 Architecture Reconstruction

Unfortunately, not all software systems have up-to-date and accurate architectural documentation. Reconstructing architecture views from existing system are needed. This dissertation proposed approach uses a set of specialized extractors and dynamic tracer to support architecture reconstruction.Architecture views are represented in diagram. The extractors are used to analyze the resources of application and extract architecture level data. The tracer is used to trace information about a software system's execution. The extracted and traced data can be shown in diagram individually, or be merged to generate architecture diagram.

In order to reduce the complexity of the architectural diagrams, clustering is used. We therefore propose a semi-automated approach. Figure 1 shows the steps involved in abstracting out the architecture views.

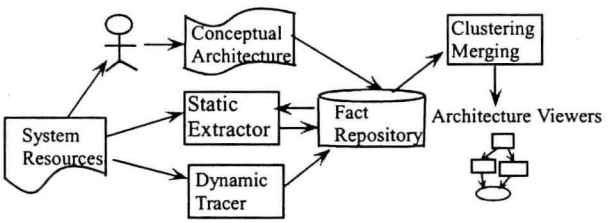


Figure 1. Reconstruct Approach for Views

Through a case study, shows how the architecture views aiding to gain a better understanding of a concrete web application system and how to assist in it's maintenance. Figure 2 is the componet architecture view of E-Process system.

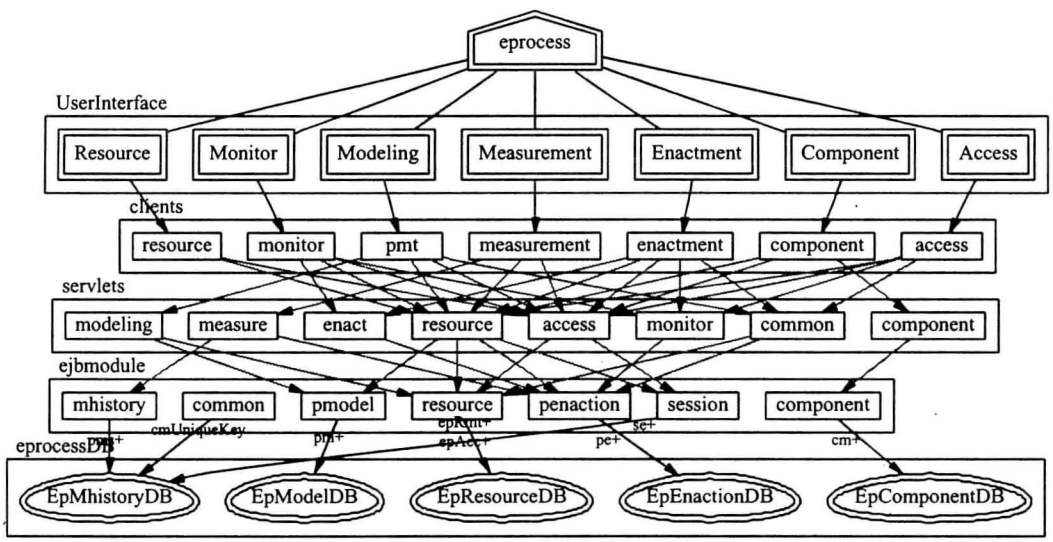


Figure 2. Component Architecture View

A Technique for Extracting EJB Components from Servlets

DONG KWAN KIM, YOUNG JONG YANG, HYO TAEG JUNG

S/W •Contents Technology Department ETRI-Computer & Software Technology Lab.,
161 Kajong-Dong, Yusong-Gu, Taejon, 305-350, KOREA
{dgkim, yangyj, htjung}@etri.re.kr

ABSTRACT

The software reuse technology as the distinguished theme of the software engineering has been studying over the several decades. It covers from simple copy and paste to software component technology. A variety of concepts and methods for the software reuse has been developed and used to make software. It is usually classified according to either what are reused targets or outputs of which step in the software development lifecycle are reused.

In this paper, we focus on the software reuse technology for Web-based applications. We also use the software component technology. In other words, we are trying to use the software component for reusing the Web applications. A lot of Web applications was developed and has been developing. Moreover, the existing systems that have been run in the legacy-computing environment are transforming into the Web environment. The purpose of our paper is to extract reusable components from the established Web applications and to help software developers make Web applications more easily. More specifically, we present the technique to transform the servlet - based Web applications into the Enterprise JavaBeans (EJB)-based ones and show a prototype tool. Our tool consists of servlet code analyzer, diagram viewer, EJB component extractor, EJB component generator, and EJB component deployer.

KEYWORDS

software reuse, Enterprise JavaBeans(EJB), servlet, software component, entity bean, session bean

1. PROCESS OF COMPONENT EXTRACTION

In this section, the analysis and design information of our proposed S2EJB tool is described. We chiefly present the component extractor of our tool.

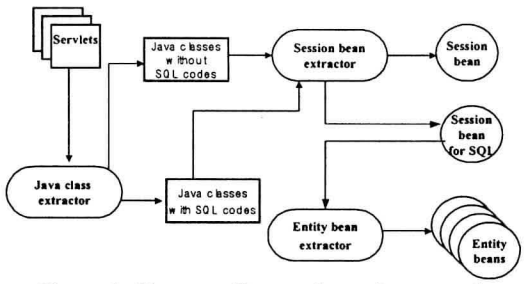


Figure 1. The overall procedure of proposed component extraction

In Figure 1, the component extractor mainly consists of Java class extractor, session bean extractor, and entity bean extractor. The Java class extraction module classifies classes considering that they have Structured Query Language(SQL) codes. Entity beans represent business data in the database and add behaviors specific to that data and session beans don't represent data like entity beans. They represent business processes or logics that perform a service such as registering, canceling, or dropping a course. Therefore we concentrate on SQL codes. If Java classes have SQL codes, they are classified as candidates for entity beans. Otherwise, they are considered as candidates for session beans. In the next step, the session bean extractor transforms classes without Java classes into session beans. Of course, we need to make sure that the classes do not include any code that is related with graphical user interface (GUI) using Abstract Window Toolkit (AWT) or SWING components. It consists of the session bean information creator and session bean mapping manager. We can make session and entity beans from the Java classes with SQL codes. The SQL codes in the Java classes are based to create the entity beans. The entity bean extractor identifies information for entity beans from session beans that have SQL codes. It consists of the entity bean information creator and entity bean mapping manager.

2. PROTOTYPE TOOL : S2EJB

2.1. Each module of S2EJB

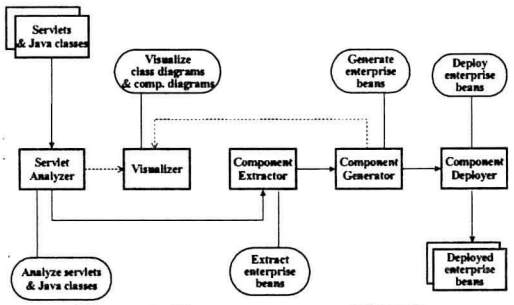


Figure 2. The component of S2EJB

S2EJB in Figure 2 consists of servlet analyzer, visualizer, component extractor, component generator, and component deployer. The rectangle in Figure 2 indicates the module of S2EJB and the round rectangle shows the functionality of each module. The dashed lines in Figure 2 draw the input flow to the visualizer. The input of the tool is servlets and Java classes and the output is deployed enterprise beans. Servlets and Java classes are given to the servlet analyzer that analyzes them. The analyzed information is used at the component extractor that extracts the enterprise bean. The component generator generates the enterprise bean by using the extraction information. The component deployer deploys the generated component at the running EJB server. In addition we can view the diagram such as class diagrams and component diagrams through the visualizer.

The servlet analyzer takes the servlets and Java classes and creates the meta information that is used when the component extractor is extracting EJB components. It also analyzes SQL statements such as select, update, delete, or insert, that are included in the Java classes and servlets as well. The analyzed information is taken over the component extractor and then the component extractor performs the process of the extraction by using the information.

The visualizer provides two kinds of diagram. One is a class diagram and the other is a component diagram. Each diagram follows the notations of the UML. The class diagram visually represents the classes that are inputted to the servlet analyzer and their relationships. The operations and attributes of the classes can also be shown. The component diagram shows the components that are generated by the tool and their relationships.

The EJB component extractor can extract both the entity and session bean. First of all, we describe the extraction procedure of the entity bean. We focus on the database table that the given Java classes or servlets access because the entity bean corresponds to a row in the database table. The extraction of the entity bean is done by analyzing the SQL statements that are included at the Java classes or servlets. Unlike the session bean, the extraction process of the entity bean includes the step to extract finder methods. So it needs to get the meta information for the finder methods. The analyzed SQL commands are mapped into the methods of the entity bean as the table 1.

Table 1. The mapping of between SQL commands and entity bean methods

SQL command	the method of entity bean
select	finder method
insert	create(), ejbCreate()
Update	ejbStore()
delete	remove(), ejbRemove()

The extraction process of the session bean is considered as the part of one of the entity bean.

The source codes of the home interface, remote interface, and bean class are generated at the component generator. In the case of the entity bean, the source code of the primary key also is generated. The source codes that are generated at the EJB component generator are compiled and then are inputted into the EJB component deployer. It deploys the generated components at the running EJB server.

3. CONCLUSIONS AND FUTURE WORKS

In this paper we have presented a technique for extracting EJB components from web applications written in servlet. We think that web applications usually have reusable modules. Therefore, we can make them reusable components such as session beans or entity beans of EJB. Afterward application developers are able to build some applications by using these components. Our proposed technique can be summarized in the following. First of all, we analyze web applications that consist of Java classes and servlet. The main role of the analysis process is to classify classes according to SQL codes. Therefore each class is placed in either a class group that has SQL code or not. Simply, classes that have SQL codes are transformed into entity beans and the other classes are converted into session beans. Specially, in using EJB component model we are sure of platform and DBMS independency and deploy generated components at various environments by modifying deployment descriptors. We need to study a clustering technique for servlets and Java classes that have the dependency relationship each other. In addition, we are investigating further techniques for automatically extracting business methods within servlets or Java classes.

Modeling and Safety Analysis of Moving Block Railway Interlocking System

Nazir A. Zafar and Keijiro Araki
Department of Computer Science and Communication Engineering,
Kyushu University,
Hakozaki 6-10-1, Higashi-ku, Fukuoka 812-8581, JAPAN
Tel : +81 92-642-3865, Fax : +81 92-642-3865,
{nazafar,araki}@dontaku.csce.kyushu-u.ac.jp

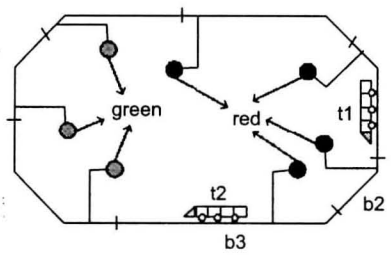


Figure 1: Fixed Block Interlocking System

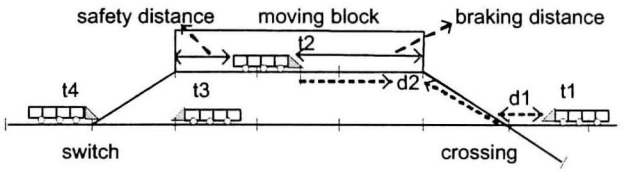


Figure 2: Moving Block Interlocking System

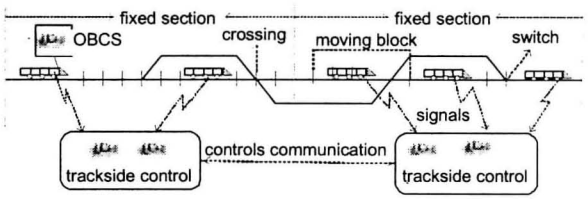


Figure 3: Concept Diagram of the Proposed Model

SUMMARY

BACKGROUND

The use of computers in safety critical systems (SCS) e.g. railway interlocking system (RIS) has increased the concern for safety. Formal methods increase the quality and provide highest confidence in this area.

OBJECTIVE

There are three major objectives to be achieved: (i) applying formal approaches in SCS, (ii) modeling moving block RIS and (iii) safety analysis preventing collision and derailing.

RAILWAY INERLOCKING SYSTEM (RIS)

Task of RIS is preventing trains from collisions and derailing. A brief introduction of existing RIS technologies is given.

Fixed Block Interlocking

In fixed block RIS, the railway network is divided into blocks which are separated by signals as shown in Fig. 1. At one time, only one train can move in a block and can enter into a block only if the next is clear. In Fig. 1, the train t1 can only enter block b2 when train t2 has cleared the block b3.

Moving Block Interlocking

The concept of moving block RIS is based on keeping only safe distance between trains. Instead of cutting piece of line into fixed blocks, the train's occupying area, some distance in front and back of train becomes the moving block. Consider Fig. 2, the trains t1 and t2 are approaching the crossing. The distances of trains t1 and t2 from crossing are represented by d1 and d2 respectively. Since, d1 is less than d2 and hence the crossing will be reserved for train t1 according to supposition in the model. In real railways, there might be some other priorities in reserving the crossing.

PROPOSED MODEL

The concept diagram is presented in Fig. 3. The railway network is divided into fixed sections controlled by trackside controls. A train is equipped with computer based On Board Control System (OBCS) for controlling train and communicating with trackside controls. The system's concept does not require signals along the tracks.

Formal Specification (only static part is presented here)

Real Topology: Real topology is composed of linear track segments (track segments) as shown in Fig. 4. Every track segment is connected with at most three segments.

Model Topology: Track segment in real topology is represented by node in model topology as in Fig. 5. An edge in the model topology is a result of connecting two track segments

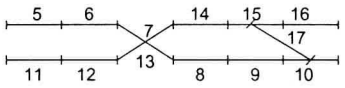


Figure 4: Real Topology

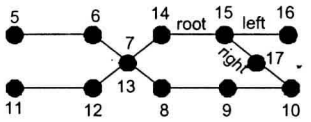


Figure 5: Model Topology

to allow a train to move to and from. Model topology is composed of tracks, switches and crossings with given details.

Tracks: Track segment is described by `Track` and edge by `Connected` in VDM-SL. The ordered pair (s1,s2) is one of the values of `Connected` and its inclusion in model topology means that train can move from track segment s1 to s2. Tracks = { (5,6), (6,5), (6,7), (7,6), (7,8), (8,7), (8,9), (9,8), (9,10), (10,9), (11,12), (12,11), (12,13), (13,12), (13,14), (14,13), (14,15), (15,14), (15,16), (16,15), (15,17), (17,15), (17,10), (10,17) } is an example of `Tracks` as shown in Fig. 5.

```
1. Tracks = set of Connected
 .1 inv tks == forall mk_(t1,t2) in set tks &
 .2 t1 <> t2;
2. Connected = Track * Track;
3. Track = token;
```

Switches: A switch is consisting of a root, left branch and right branch as shown in Fig. 5. The left and right branches are designated from root of a switch. Switch is specified as `Switch` with three components `root`, `associate` and `control` in VDM-SL. For example, `root = (15,14)`, `associate = {(15,14),(15,16),(15,17)}` and `control = {15,16) |-> <LEFT>,(15,17) |-> <RIGHT>}` as in Fig. 5. Formal specification is given below.

```
4. Switches = map Track to Switch
 .1 inv sws == forall s in set dom sws &
 .2 forall mk_(s1,s2) in set sws(s).associate &
 .3 s = s1 ;
5. Switch :: root : Connected
 .1 associate : set of Connected
 .2 control : map Connected to SControl
 .3 inv swt == card swt.associate = 3 and
 .4 swt.associate = dom swt.control union
    {swt.root};
6. SControl = <LEFT>|<RIGHT>;
```

Crossings: The crossing is modeled by two crossovers where crossover is a set of adjacent edges as in Fig. 5. For example,

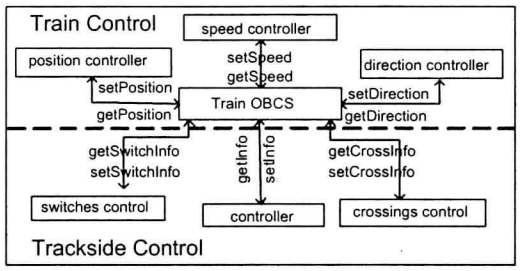


Figure 6: Formal Model

{(6,7),(7,6),(7,8),(8,7) } and {(12,13),(13,12),(13,14),(14,13) } are two crossovers formulating a crossing. `Crossings` is the mapping from `Track * Track` to `Crossing`. An ordered pair (7,13) is one of the element in domain of the mapping. It is to be noted that any element in the domain of `Crossings` is not an edge in the model topology.

```
7. Crossings = map (Track * Track) to Crossing
 .1 inv cs == forall mk_(s1,s2) in set dom cs &
 .2 forall mk_(s3,s4) in set cs(mk_(s1,s2)).
    crossover1 & s1 = s3 or s1 = s4 and
 .3 forall mk_(s3,s4) in set cs(mk_(s1,s2)).
    crossover2 & s2 = s3 or s2 = s4;
8. Crossing :: crossover1 : set of Connected
 .1 crossover2 : set of Connected
 .2 inv cross == card cross.crossover1 = 4 and
 .3 card cross.crossover2 = 4 ;
```

Formal Model

Architecture and activities of the interlocking components (trains, controls) and network components (switch, crossing) are presented as in Fig. 6. All the functions shown in Fig. 6 are self explanatory and formalized. The bi-directional arrows show both side communication and dotted line distinguishes activities of Train OBCS (one part for monitoring train and other interacting with trackside controls).

RESULTS

Following are the main steps involved in the work.

- Topology is proposed supporting moving block RIS.
- Formal specification (static and dynamic) of components is described in such a way that it will ease further refinement and modification. Only static part is given in this summary.
- A part of model which shows interaction of network and interlocking components is represented. It was helpful in understanding functionality of RIS components. Formal definitions of some of the functions are presented to give an overview of the model.
- Trackside controls are used replacing physical signals along tracks. Position of train can also be recorded by the controls in addition to other functionalities e.g. observing trains, controlling switches and crossings, allocating routes etc.
- The abstract safety requirements, no collision and no derailing, are analyzed and formal specification is described.

Digital Watermarking Algorithm Using Extraction Of Contours

Doo-Hyun Kim, Yong-Sung Kim, OK-Chang , Cheol-Jung Yoo
Chonbuk National University, Chonju, Republic of Korea
kdh@mail.chonbuk.ac.kr {yskim, okjang, cjyoo}@moak.chonbuk.ac.kr

In this paper, digital watermarking algorithm that is able to protect the copyright without referring to the original image and watermark using counters detection is proposed. Without referring to original image, we find location of watermark embedded using extraction of contours for the image. Then watermark will embed where found. Also, without referring to watermark, we use binary code by information of owner that is unique each owner. It is benefit to be able to judgment copyright of owner by only watermarked image. Watermark embedding method is distinguished that two groups are chosen by the watermark. A software simulation shows that the proposed method assures the above advantages of the retrieved watermark.

1. INTRODUCTION

Because of rapid improvements in digital technology and information can obtain a lot of information on the Internet. However, there are some problems using information. According to increase commercial value of digital data, it was used illegally. Therefore, an important issue for electronic publishing is copyright protection.

Since digital watermarking techniques can be used to prove the copyright ownership of digital data when an illegal copy is found, such techniques have received the attention of many researchers around the world. By securely embedding a small amount of information into a digital data, only the person who knows the secret can prove that such hidden information exists.

Generally, the main requirements that any watermarking technique should satisfy can be briefly described. As the watermark is embedded into the original image, the distortion shouldn't be recognizable by the eyes. Also, when the watermarked images are compressed and retrieved by lossy compression such as JPEG the watermark should still be recognizable after extraction.

There are two common methods of watermarking, namely spatial and frequency domains.

In the spatial domain watermarking methods, the intensity or amplitude of the image pixel is directly modified by a small amount by the watermark data, undetectable to the human eye. This technique can do easily watermark embedding process. However, when some common signal processing method is applied to the watermarked image, the watermark signal is easily destroyed.

Recent the frequency domain watermarking methods are based on the discrete cosine transform(DCT) are added to the DCT coefficients at the middle frequencies as signatures. This technique isn't easily destroyed.

However, when watermark detection processing compares original image with watermarked image, it is complicated algorithm. Also, it needs original image and watermark to prove the copyright ownership of digital data.

In this paper, digital watermarking algorithm that is able to protect the copyright without referring to the original image and watermark using counters detection is proposed.

Without referring to original image, we find location watermark embedded using extraction of contours for the image. Then watermark will embed where found. Without referring to watermark, we use binary code by information of owner that is unique each owner. It is benefit to be able to judgment copyright of owner by only watermarked image. Watermark embedding method is distinguished that two groups are chosen by the watermark. Watermark can extract to watermarked image by watermark detecting algorithm.

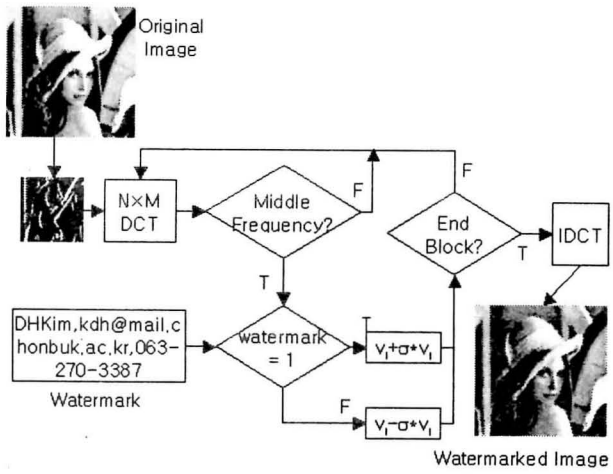


Fig. 2. Watermark embedding block diagram

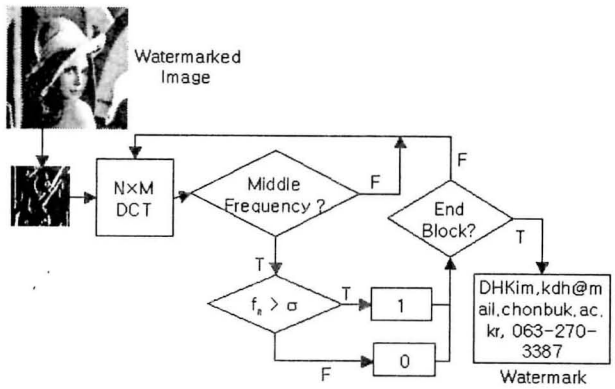


Fig. 3. Watermark extracting block diagram

Fig. 4. Result of watermark generating

Fig. 5 original images

Noise reduction Contours detection Watermarked image

Fig. 6 The Result of proposed watermarking processing

Fig. 7. Retrieved watermarks

Automatic Flowchart Layout for Program Visualization

Wei Lai* and **Mao Lin Huang**+

*School of Information Technology
Swinburne University of Technology
POBox 218, Hawthorn, Vic 3122, Australia
wlai@it.swin.edu.au

+Faculty of Information technology
University of Technology, Sydney
Sydney, Australia
maolin@it.uts.edu.au

Summary

Diagrams associated with programs (such as data flow call graphs, flowcharts, and Nassi-Shneiderman diagrams) help programmers or users to understand what the programs do, how they work and why they work. Flowcharts have been used in programming since the birth of computers. It would be very helpful for understanding large software projects and maintaining them by converting program source code to flowcharts. For instance, when changes are made to source code, corresponding changes will be shown by related flowcharts.

Several systems have been developed to visualize programs as flowcharts. However, automatic layout and effective management of flowcharts are critical for software visualization systems. The current approach is that the graphical appearance of a flowchart is pre-defined by the user or designer, that is, one needs to specify the geometric information in detail, such as shape, size and location of every node image in a flowchart. This is a tedious task, especially for converting a large program to a flowchart. It is necessary to investigate on automatic layout and efficient management of flowcharts.

Here we present an approach to designing visual flowchart components. This approach not only can represent all flowcharts but also support the construction and automatic layout of flowchart components.

Flowchart components should be designed to represent structured programs. A program is a set of flowchart components. The flowchart components are shown in Figure 1. Figure 2 shows a Pascal program and the process of displaying it hierarchically.

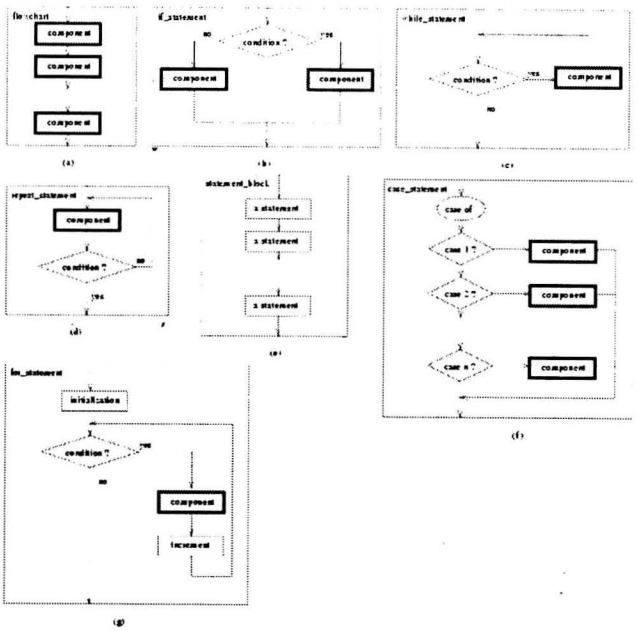


Figure 1: The flowchart components

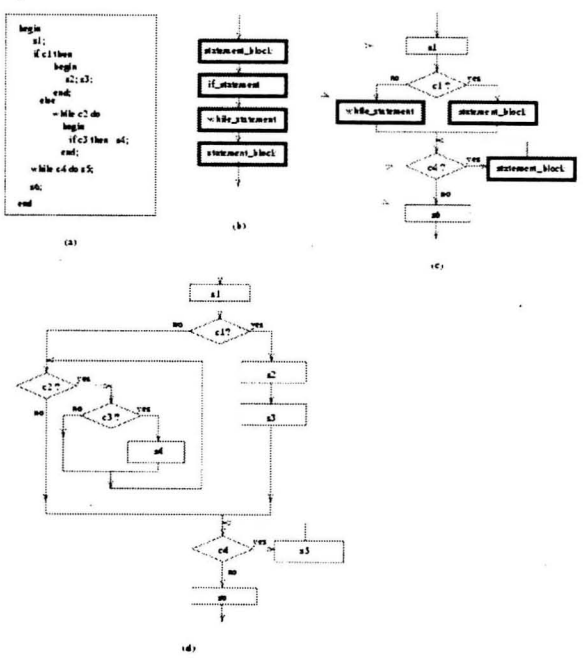


Figure 2: A Pascal program and its corresponding flowcharts

Progressive Data Integration Based on Data Visibility

Dong Won Jeong **Doo-Kwon Baik**

Software System Laboratory, Korea University

1, 5-ka, Anam-dong, Sungbuk-ku, Seoul 136-701, Korea

withimp@software.korea.ac.kr baik@software.korea.ac.kr

MAIN IDEA

Our key idea is very simple. There is no need to integrate all data in target databases at once. For example, some users are just interested in simple and easy information. But some users such as experts are interested in more complicated and specialized data including common data too. So, all data can be divided into several views hierarchically by the data visibility. Therefore we can build metadata registries hierarchically. Also we can gradually expand the metadata registry of the highest level in bottom-up approach. Data visibility is related to data specialization, user specialization and data quantity. Figure 1 describes well our research's main idea.

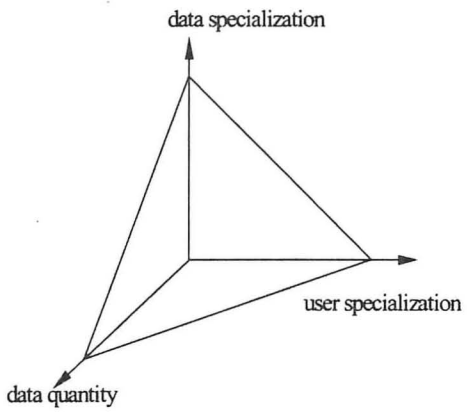


Figure 1. The conceptual diagram of the data visibility

CONCEPTUAL MODEL

The LoG method consists of Interface layer, Global MDR layer, Local MDR layer and Data resource layer. This classification is based on the user level from data visibility handled in the section 3.2. Figure 3

shows the LoG methodology conceptually.

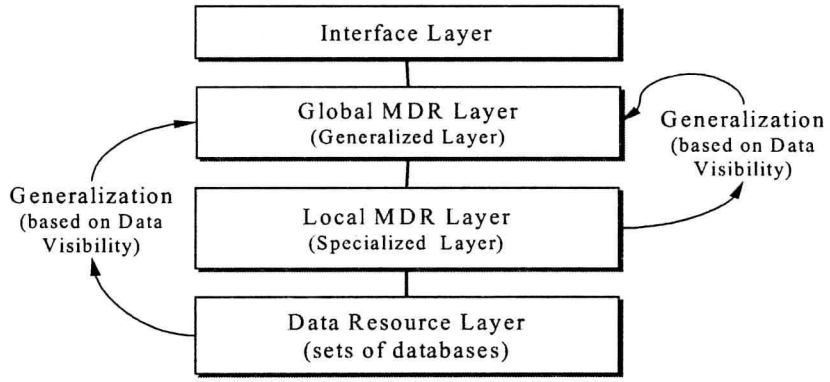


Figure 2. The conceptual model of the LoG methodology

In Figure 2, the highest layer is the interface layer that consists of two sublayers- local interface layer and global interface layer. The interface layer provides such services as searching, querying and viewing that users access to send some queries and receive the results. The second layer is the global MDR layer that manages the generalized metadata registry and repository that support the general users in high-level to use data as a whole. The local MDR layer, the third layer, supports experts that use more complicated and detailed data. The users in this layer communicate with a LoG method-based system through the global interface layer that is one of the sublayers of the interface layer. This layer manages the localized metadata registries and repositories. The final layer is the data resource layer. It is the set of many databases that are created respectively and independently by each domain experts or DBAs.

We should make the metadata registries evolve to extend the guidelines whenever a new standard, i.e., a metadata element, is generalized. That is, we must reflect the changed situations to the metadata registries for the gradual and progressive integration of all data. The LoG methodology supports a mechanism to extend the existing metadata registries progressively.

The generalized metadata element can be created from the databases in the data resource layer. As we described already, each metadata registry can be replaced in the local MDR layer dependently and locally. Therefore a new metadata element can be created, and can be registered in the corresponding local metadata registries respectively. Also, new metadata elements are created from metadata registries in the local MDR layer, and then they can be registered newly into the global MDR in the global MDR layer.

Evolutionary Prototyping Technique Using Abstract Interpretation in Java

Hiroyuki Ozaki **Katsuhiko Gondow** **Takuya Katayama**
School of Information Science
Japan Advanced Institute of Science and Technology
1-1, Asahidai, Tatsunokuchi, Ishikawa, 923-1292, JAPAN
{ozaki, gondow, katayama}@jaist.ac.jp

INTRODUCTION

In this paper, we propose a new theoretical framework for evolutionary prototyping in Java using abstract interpretation, which allows programmers to execute a system as a whole, even though some parts of the system are abstracted or partially implemented.

Object-oriented methodologies and programming languages are widely recognized as effective tools to efficiently develop large-scale software systems. Using these technologies, systems can be designed and developed as a collection of encapsulated classes, and they can be organized as a class hierarchies. Well-designed classes result in high productivity and maintainability.

Unfortunately, however, it is often the case that we know classes are not well-designed in the very late stage of development, where all classes are thoroughly implemented. To reduce the development cost, it is indispensable to know it in the early stage. The technique of abstract interpretation can be a good theoretical groundwork to solve this problem, since abstract interpretation allows us to execute a system as a whole, even though some parts of the system are abstracted or partially implemented.

Abstract interpretation[5] is a framework for program analysis. A program is analyzed by approximated computation defined on abstract values (e.g., a positive number) and corresponding abstract operations. In contrast, our basic idea is that:

- First, we develop a system as a whole as relatively abstracted one. Then, we gradually make it more concrete, while repeating decomposition, specialization, refinement, refactoring and so on.
- Each stage in the above development, we can execute the system as a whole using abstract interpretation to make sure, even in the early stage, whether classes and methods are well designed or not.

Applying abstract interpretation to software engineering is nothing new. ISDR (Incremental Software development method based on Data Reification)[4] introduced by Yoshioka deals with function refinement in ML where programs can be executed using abstract interpretation. ISDR is based on stepwise refinement such as Refinement Calculus[1]. Refinement Calculus is a formalization of stepwise refinement to develop programs from specifications written in natural languages. In Refinement Calculus, although an intermediate program contains a part of codes, we cannot execute the program as a whole, since the program includes fragments of specifications written in natural languages.

The drawback of the previous work is that they are based on functional programming. Thus, they cannot cope with side-effects, flow of controls, variables, dynamic bindings, inheritance and so on. It is very hard to build a framework for abstract interpretation with all these issues at once on object-oriented programming languages. So, as a first step, we try to build a framework with side-effects using CLASSICJAVA[3], which is one of Java formalizations. CLASSICJAVA is a good theoretical groundwork to incorporate side-effects to abstract interpretation, because CLASSICJAVA preserves the essence of Java while being simplified.

This work can be regarded as an instance of a theoretical framework of software evolution[6]. The framework is proposed using evolution relation which assumes a mathematical lattice. Our refinement relation among objects is one of evolution relations.

Our work is similar to symbolic execution[2] in the sense that programs can be executed without using concrete values. In symbolic execution, programs are executed by dealing with variables symbolically. In contrast, programs can be executed using abstracted values in our approach.

FORMALIZATION

We define *object refinement relation* $\approx$ among objects for abstract interpretation to continue execution in run-time. We write

$$obj^{\#}@\mathcal{F}^{\#} \approx obj@\mathcal{F}$$

if the object obj with the field environment $\mathcal{F}$ can be abstracted to the object $obj^{\#}$ with the field environment $\mathcal{F}^{\#}$. $\mathcal{F}^{\#}$ and $\mathcal{F}$ represent current states of the objects. To define the relation, we introduce field environment refinement relation $\sqsubseteq^{\mathcal{F}}$ and class refinement relation $\sqsubseteq^{C}$. The field environment refinement relation expresses abstracted states of an object and their refinement, and The class refinement re-

lation expresses abstracted functions of an object and their refinement. The class refinement relation consists of *method refinement relation* $\sqsubseteq^{\mathcal{M}}$, *choice method decomposition relation* $\sqsubseteq^{\mathcal{M}}_{+}$, *sequence method decomposition relation* $\sqsubseteq^{\mathcal{M}}_{\times}$. Intuitively, the method refinement relation expresses that an abstracted method is changed to another method, which receives and returns more concrete value than the abstracted method. The choice method decomposition relation expresses that an abstracted method is decomposed into methods, where each decomposed method receives a part of values of the abstracted method and returns a part of values of the abstracted method. The sequence method decomposition relation expresses that an abstracted method is decomposed into methods, the sequence of which realizes the function of the method.

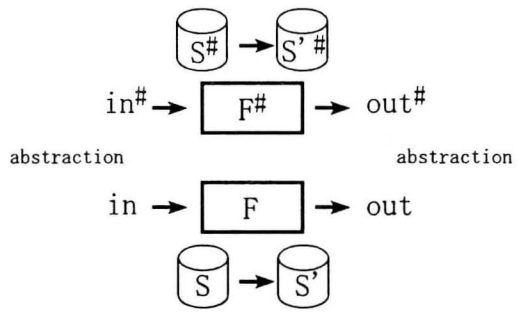


Figure 1: Concept of method refinement and decomposition

These refinement and decomposition are suitable to understand the concepts of our formalization of abstract interpretation. Fig. 1 shows these relations, where $F^{\#}$ and F represent functions consisting of a method, a choice of methods and a sequence of methods. $S^{\#}$, $S'^{\#}$, S and S' represent stores. We define both of the refinement and decomposition to satisfy that output value of $F^{\#}$ is equals to an abstracted value of output value obtained by F, where input value of $F^{\#}$ is the abstracted value of input value of F. These refinement and decomposition are defined using data domain and store refinement relation.

The data domain expresses an abstracted value and its refinement. For example, the relationship between the input value $in^{\#}$ and in, and between the output value $out^{\#}$ and out are represented in Fig. 1.

The store refinement relation expresses an abstracted store and its refinement. Since output value depends not only on input value but also on a store, the relation is needed to define the relationship between methods in terms of their behavior.

CONCLUSION AND FUTURE WORKS

We formalized the refinement relation among objects with side-effects. Using abstract interpretation with the relation, systems can be executed as a whole in intermediate stages of software development. Abstract interpretation became almost possible with this formalization. What we lack is some algorithm to decide an abstracted value and field environment uniquely, since we suppose that data domains form partial order relations. Furthermore, we will introduce class inheritance in our framework. An extension of our framework may need a refinement relation on an inheritance relation of classes. We did not deal with class inheritance in this paper, since our formalization becomes more complex. However, our framework is an important step to deal with properties of object-oriented languages such as inheritance.

REFERENCES

1. Carroll Morgan: Programming from Specifications Second Edition. Prentice Hall, 1994.
2. James C. King: Symbolic Execution and Program Testing. Communications of the ACM, Vol. 19 Issue 7, July 1976.
3. Matthew Flatt, Shriram Krishnamurthi and Matthrias Fellenisen: Classes and Mixins. In Proc. 25th ACM Symposium on Principles of Programming Languages, January 1998, pp.171-183.
4. Nobukazu Yoshioka, Masato Suzuki and Takuya Katayama: Incremental Software Development Method Based on Abstract Interpretation. Proc. of IWSSD-10, IEEE, 1998.
5. P. Cousot and R. Cousot: Abstract Interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In Proc. 4th ACM Symposium on Principle of Programming Languages, January 1997.
6. Takuya Katayama: A Theoretical Framework of Software Evolution. Proc. of Int. Workshop on Principles of Software Evolution(IWPSE98), 1998, pp.1-5.

Specification Technique of EJB-Based Application Using Design by Contracts Approach

Hye-Min Noh, Hyong-Jin Park, Cheol-Jung Yoo, Yong-Sung Kim, and Ok-Bae Chang
Depart of Computer Science, Chonbuk National University, Korea
{hmino, hyjpark}@cs.chonbuk.ac.kr {cjyoo, yskim, okjang}@moak.chonbuk.ac.kr

1 Introduction

The purpose of this paper is building reliable software, which has correctness and robustness, by proposing the specification technique for applying DbC approach to building EJB-based application.

2 DbC Approach

An effective way to specify class and operations is to use pre-condition, post-condition and invariant. Specification method using these pre-condition, post-condition and invariant is called DbC approach.

3 Specification Technique of Ejb-based Application Using DbC Approach

3.1 Basic Concept

All specifying rules that propose based on the following assumptions.

(assumption 1) Graphic model which becomes base of specification is called EJB component class diagram.
(assumption 2) Specification target of pre-condition, post-condition and invariant limits to bean class of enterprise beans.

3.3 EJB component class diagram drawing rules

We propose to draw EJB component class diagram in two forms; EJB component class overview diagram and EJB component detail diagram.

3.3.1 EJB component class overview diagram drawing rules

[rule 1] EJB component class overview diagram focuses on association between enterprise beans rather than detail information of enterprise beans. figure 1 is showing this.

3.3.2 EJB component class detail diagram drawing rules

EJB component class detail diagram drawing rules is based on the following assumptions.
(assumption 3) Session bean class model is consisted of home interface, remote interface, and bean class regarding to the association and the multiplicity, and the operation defined in the remote interface during the drawing of the model, will be omitted since it is identical to the operation in bean class.

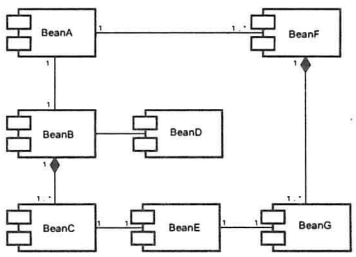


figure 1 EJB component class overview diagram

(assumption 4) Entity bean class model is consisted of home interface, remote interface, bean class, primary key class, and database table regarding to the association and multiplicity and the operation defined in the remote interface during the drawing of the model, will be omitted since it is identical to the operation in bean class.
(assumption 5) The table of the database associated with the entity bean class model will be regarded as an class and will be added to the bean class model and the records of database will be regarded as an instance of the table.

<definition 1> The bean class has an association with the home /remote interface and the multiplicity is 1 : 1.
<definition 2> The bean class has an association with the primary key class and the multiplicity is 1 : 1
<definition 3> There is an association between the bean class and the database record in the entity bean class model and the multiplicity is 1 : 1...*

The drawing rule of the EJB component class detail diagram applying the above assumptions and definitions are as following.

[rule 2] Session bean class model will be written in a form of figure 2 according to *(assumption 3)* and *<definition 1>*.

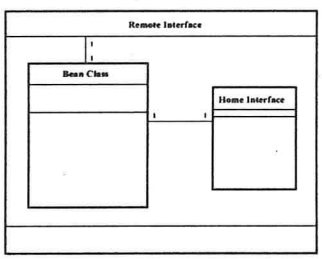


figure 2 Session bean class model

[rule 3] Entity bean class model is drawn in the same form as figure 3 through *(assumption 4), (assumption 5), <definition 1>, <definition 2>,* and *<definition 3>.*

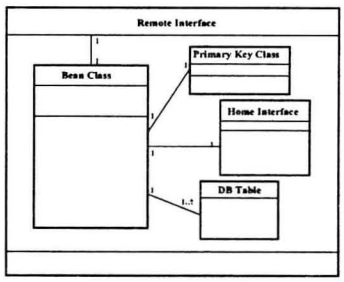


figure 3 Entity bean class model

[rule 4] EJB component class detail diagram using the enterprise beans class model which is drawn by *[rule 2]* and *[rule 3]* is drawn in the same from as figure 4.

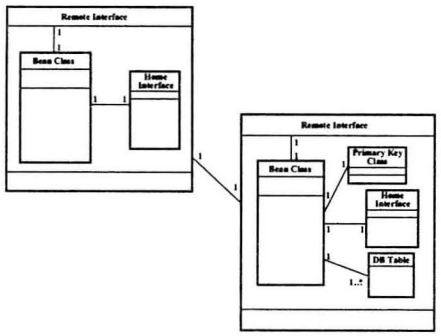


figure 4 EJB component class detail diagram

3.4 Invariant Writing Rule

<definition 4> Invariant shows the conditions of the bean class to be shown in the OCL expression which has to be constantly maintained.

Invariant is to be written according to *[rule 5]* and *[rule 6].*

[rule 5] Invariant of bean class is to be written as the following form.

context BeanClassName inv:
OCL expression

[rule 6] The OCL expression which consists the invariant, pre-condition and post-condition is to be written according to the OCL type and grammar defined in the OCL specification.

3.5 Pre-condition and Post-condition Writing Rules

Pre-condition and post-condition are drawn according to the general OCL writing rules. So no assumptions are necessary. The following is defining the significance of pre-condition and post-condition and types of operation in bean class in the EJB-based application.

<definition 5> Pre-condition shows the conditions that are needed for each of the operations in bean class to be executed.

<definition 6> When pre-condition is satisfied, post-condition shows the services which each of the operations of bean class has to provide.

<definition 7> Operations of bean class is consisted of the EJB Required operation which is called by the container of application server and general business operation.

[rule 7] Pre-condition and Post-condition about the operation of bean class is to be written the following format.

```
context BeanClassName::operationName(param1 : Type1, ...):
ReturnType
  pre {precondition name} : OCL expression
  post {postcondition name} : OCL expression
```

[rule 6] The OCL expression which consists invariant, pre-condition and post-condition is to be written according to the OCL type and grammar defined in OCL specification.

3.6 Specification Phase

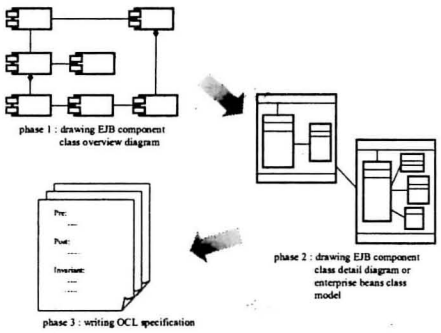


figure 5 OCL specification writing phase of EJB-based application

4 Meaning of specification

The specification technique written on this paper gives three types of significance to these participants.

- Significance as a software design method
- Significance as software document
- Significance as the basis to bring out the test case of the software

A Formal Refinement Method of Statechart

Shusaku Iida
School of Network and Information
Senshu University
2-1-1 Higashimita, Tama,
Kawasaki, Kanagawa, Japan
iida@isc.senshu-u.ac.jp

Kokichi Futatsugi
Graduate School of Information Scicence
Japan Advanced Institute of Science and Techinology
1-1 Asahidai, Nomi, Ishikawa, Japan
kokichi@jaist.ac.jp

INTRODUCTION

Statechart in UML has a rich expression power. However, using Statechart in system developments is not so easy. Because, deriving all the possible states from the system requirement is sometimes very hard. Moreover, one should select a proper abstraction level which she/he needed for the stage of system developments. One classical solution for this problem is to use a refinement based development technique. Firstly, specify a highly abstract Statechart which only includes simple and essential states and use stepwise refinement to get more concrete one.

In this article, we propose a formal refinement method of Statechart. We especially focus on automation of refinement verifications. Our final goal is to build a CASE tool which support automated refinement proof. To support Statechart refinement mechanically, we have to provide formal semantics of Statechart and then define refinement in Statechart formally.

SYSTEM DEVELOPMENTS WITH STATECHART

FSTD is commonly used in embedded system developments. Embedded system is usually a system, which involves both software system and hardware system. This software system is implemented to the system to controle the hardware. The hardware systems (e.g. CD player, air conditioning system, cruise control system, etc) usually have states like "playing", "ejecting", "stop" and so on in the case of CD player. However, in ordinary software systems, the states of a system are not so clear compare to embedded systems. This might be one reason that FSTD is more commonly used in embedded system developments (or hardware based system developments) rather than ordinary software system developments.

When designing embedded systems, information from hardware systems are important. A typical embedded system receives information from a hardware system and sends back signals to control it. This information from hardware systems can be regarded as a result of observations of the hardware systems and they can be represented as variables in a Statechart diagram. For example, air conditioning system has to know the current temperature of the room and the target temperature.

Variables are playing important role in system developments with Statechart. Some CASE tools used in embedded system developments (for example, ZIPC) adopt FSTD as its designing language. According to the success of UML, these tools begin to use UML Statechart as it language. A Statechart diagram (or FSTD) can translate to a real programming code if it has enough information (ZIPC has such function). So, system designer starts from the simple and essencial Statechart diagram built from the requirement and gradually refine it by adding new variables and states until it contains enough information.

The key point in this development method is refinement. Refinement is a relation between two Statechart diagrams. A refined Statechart contains more information compare to the original one but preserving its structure. To ensure that a refinement relation is proper in a CASE tool, we need a formal semantics of Statechart and formal definition of refinement relation.

A SEMANTICS OF STATECHART

To simplify the discussion, we have restricted the syntax of Statechart as follows:

- events and actions are represented in OCL,
- guards in events are only constructed by Boolean operations,
- only a variable assignment is allowed in an action,
- there should be an initial state (also in all the sub-states),
- and we don't consider concurrency.

Some of these restrictions might be too strong for real system developments. However, we consider that these conditions can be easily relaxed and we are planning to do it in the near future.

We define states of a system by a set of tuples consists of a variable and its value. We call this set a "configuration" and represent it as $\{(V_1, X_1), (V_2, X_2), ..., (V_n, X_n)\}$, where n is natural number, V_n is a variable, and X_n is a value for V_n. This tuple is called an "attribute". Each attribute represent observable aspect of the system.

For example, { *(engine, on), (speed, 60)* } is a state of CAR2 specified in figure 1. Note that a state of a system may differ from a state of Statechart diagram. In Statechart diagram, a state name may represents several states of the sys-

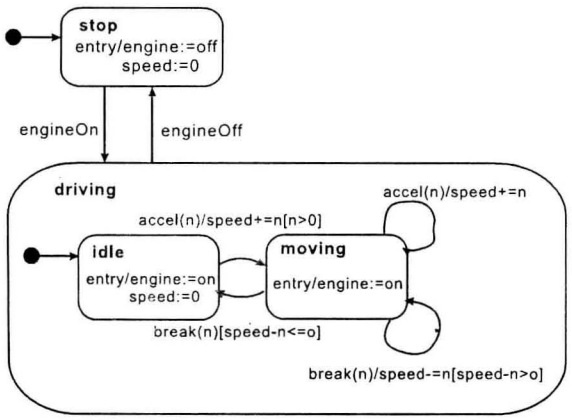


Figure 1: CAR2

tem. We define the name of a state in a Statechart diagram as a constraint for the configuration built from the diagram. For example, the state "moving" means that "engine = on" and "speed > 0". Therefore, there may be infinite numbers of states in "moving" according to the value assigned to the variable "speed".

A transition is an operation (function) on a configuration. For example, the event "accel" in figure 1 and the actions triggered by it are represented by the operation "accel" which only defined if the configuration meet the constraint "moving" or "idle". For example, if the configuration is { *(engine, on), (speed, 0)* }, *accel(n)* takes { *(engine, on), (speed, X)* } and returns { *(engine, on), (speed, X+n)* } (where $n > 0$).

A hierarchically represented state can be flattened by following rules:

- a transition from a state to its sub-state should go to the initial state of the sub-state
- a transition from a sub-state to its super-state should translated to transitions from all the sub-state to the super-state

A formal semantics of Statechart by using formal specification language

Our goal is to handle refinement proof by CASE tools. To achive this goal, we need a definition of what is refinement relation and a methodology to prove that a Statechart diagram is refined by the other one. We can build own definition base on our semantics of Statechart from the scratch; however more simple and safe way is to use an already established framework. We use algebraic specification technique. Some algebraic specification is called "executable" which means that there is a software system which supports specification based simulation, testing and verification. Therefore, if we can build an algebraic specification version of our semantics, we can immediately use it as a core engine for refinement checking in our CASE tool. We use algebraic specification language CafeOBJ to describe our semantics.

REFINEMENT

A refinement is a relation between two Statechart diagrams. We already provide a formal semantics of Statechart diagram, so we can say that refinement relations of Statechart is a refinement relation defined over two CafeOBJ specifications of state machines. Suppose we have two CafeOBJ specification α and β, β refines α means that there exists a homomorphism from α to β. So, if we want to prove Statechart B refines Statechart A then we interpret A and B to CafeOBJ specification α and β respectively and then prove a refinement relation between them.

To prove refinement relation in CafeOBJ we do the following steps:

1. define morphism ϕ between $Sig(\alpha)$ to $Sig(\beta)$ where Sig takes a specification and returns its signature [1],
2. translate equations defined in alpha by using ϕ so that the equations can be proved in β,
3. and prove the translated equations in β.

We consider two types of refinement "attribute refinement" and "state refinement". Attribute refinement is a process of adding new variables and state refinement is a process of adding new states.

CONCLUSION

Statechart has rich expression power. However, there is a difficulty that one should identify all the possible states of the system for each abstraction level of development stages. In this article, we model the state space by using configuration and provide a two refinement methodology "attribute refinement" and "state refinement". By using attribute refinement one can expand a configuration and by state refinement, one can add a constraint for a configuration, which means adding a state to Statechart diagram.

We adopt CafeOBJ for defining the semantics of Statechart diagrams and define refinement relation of Statechart. CafeOBJ is executable specification language, so that it can be used as a core engine of CASE tools which support our refinement methodology. In this article, we obtain CafeOBJ specification from Statechart diagram by hand translation; however, this can be done automatically in the near future.

[1] the signature of a specification consists of definitions of sorts and operations

Commit Mechanism of Engineering Database Supporting Cooperative Design Transaction

Guoning Chen **Taoshen Li**
College of Computer and Information Engineering
Guangxi University
Daxue road 100, Nanning 530004, P.R.China
guon_chen@163.com

Guoqiong Liao
College of Computer Sci. & Tech.
Huazhong University of Sci. & Tech.
Wuhan 430074, P.R.China

INTRODUCTION

The Application of cooperative transaction

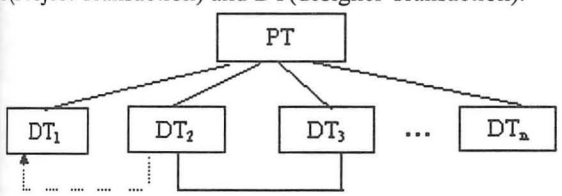
Cooperative design, including cooperative CAD
Cooperative edit/compile
Cooperative software
Workflow
Other multi-user environment

Some Research on cooperative transaction

TRANSCOOP project, SE-EDBMSII system, ConTran model, flexible cooperative transaction model
All of them do not give a satisfied solution for the cooperative transaction commit process.

TRANSACTION MODEL SUPPORTING COOPERATIVE DESIGN

The cooperative transaction model used here has just two levels.
PT(Project Transaction) and DT(designer Transaction).



cooperative read cooperative modification

- A PT is completed by two or more DTs cooperatively. (e.g., PT consists of DT_1, DT_2, ..., DT_n)
- Cooperative read------Some transactions may read the uncommitted results belonged to other transactions.
- Cooperative modification------Some transactions may modify the same design object cooperatively.

ARCHITECTURE OF TRANSACTION PROCESS

The architecture used here is based on the extended C/S. It allows the transaction access data through check out/in mechanism and traditional remote data access.
However, the transaction model can also be implement on 3-tier structure.

TIMEOUT-TRIGGER TWO-PHASE COMMIT PROTOCOL (TT-2PC)

What is Timeout-trigger Mechanism?

Timeout is the time that the transaction must finish and decide whether to commit or not.
Before timeout, the participants can apply to commit if they finish the design and the design result passes through the check of local constrains which are checked out from the public DB in the initialization. The coordinator can accept the application if the lock releasing operation of the participant completes successfully. Otherwise the coordinator will reject the application, and the participant must wait. If the application is accepted, the coordinator will judge whether the finishing condition of the coordinator transaction is held or not afterward.
When timeout comes, the protocol deals with the commit process just like basic 2PC protocol.

Why we use timeout-trigger?

We don't allow the participants to abort their transactions without the demand of the coordinator, for it may disturb the execution of the whole transaction. However, there is sometimes that participants have no choice to abort. How should we do? We delay their decision until a flag comes. The flag is timeout-trigger. When timeout-trigger comes, the coordinator will ask all unfinished participants to make a decision. Then, the abort-wanted participants may submit their decisions now.

THE PROCEDURE OF TT-2PC

Timeout flag divides our protocol into two part described above. For simplicity, we just consider the case before timeout comes, and some special operation such as log write and the strategy of regulation checking are not contained here.

Before timeout:

Coordinator:

Initialize;

(2) Wait for the commit application

.....

Receive a ' commit' application from a participant;

Use some regulation to judge whether it can commit or not

If it can commit

Send ' accept-commit' message to the participant;

Accept the results of the participant;

Check the finishing condition of the coordinator;

If it held

Send ' terminate' message to all being executed participants;

Move the committed results to public DB;

Wait for ' ack';

.... got o (2)

Else continue to wait goto (1)

Else

Send ' not accept' message to the participant;

Goto (1);

(2) If receive all ' ack'

Write log that coordinator ' commit';

Finish process;

Participants:

Receive ' tran-begin' message from coordinator;

Initialize;

Begin operation;

.....

If operations are finished and the results pass the check of regulation

Then Send 'commit-apply' message and the results to coordinator;

If receive ' accept-commit' message from coordinator

Write log that local transaction ' commit';

Commit process;

Send ' ack' message to coordinator;

Else if receive ' terminate' message from coordinator

Finish process;

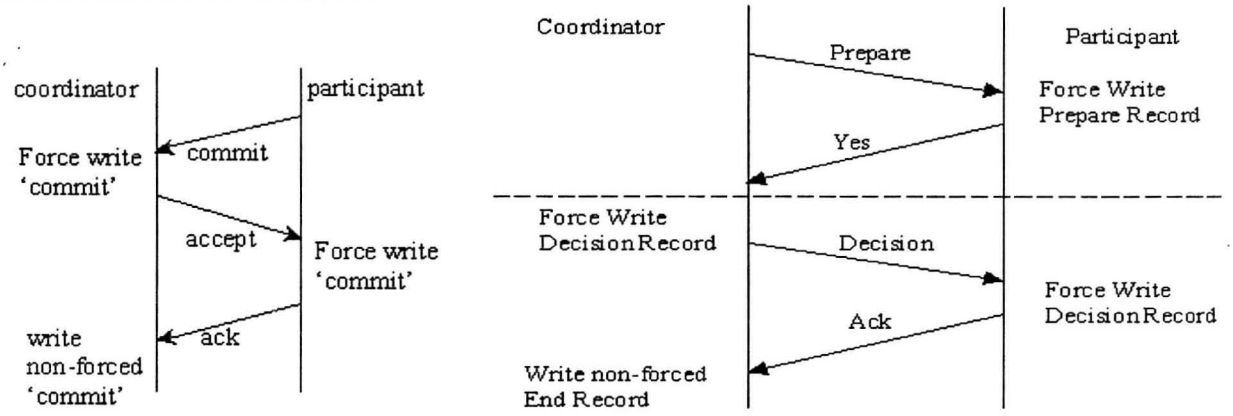
Send ' ack';

Else continue to wait;

After timeout:

The commit process is just like basic-2PC protocol

THE PERFORMANCE DISCUSSION



(a) non-timeout commit case (b) after timeout case

Message exchange and log write during the TT-2PC

The Comparison of Timeout-Trigged Commit Protocol with Basic 2PC, PrA and PrC

Type of commit protocol	TT-2PC		Basic-2PC	PrA	PrC
	After timeout	Before timeout			
Message complexity	4n	3n	4n	4n	3n*
Log complexity	2n+1	2n	2n+1	2n+1	n+2*

(a) coordinator commit case

Type of commit protocol	TT-2PC		Basic-2PC	PrA	PrC
	After timeout	Before timeout			
Message complexity	4n	-	4n	3n*	4n
Log complexity	2n+1	-	2n+1	n*	2n+1

(b) coordinator abort case

CONCLUSION

Preserving Constraints in Mapping XML DTD to Relational Schema

Yunsheng Liu, Hao Zhong, Yi Wang, Mingjun Chen
Huazhong University of Science and Technology, Wuhan, China.

ABSTRACT

In the paper, we present a middle model (LDG) to describe schema in XML DTD. Further, we propose an approach to present DTD with LDG, and the corresponding algorithm of automatically mapping LDG to relational schema.

KEYWORDS: XML, DTD, Mapping, Constraints, Schema

INTRODUCTION

XML is emerging as a possible candidate data format because it is simpler than SGML, and more powerful than HTML. One way to store XML data is to reuse the established relational database techniques by mapping and storing XML data in a relational storage.

In this paper, we use a middle model, Labeled-Directed Graph (LDG), to describe DTD schema. Our approach includes three steps. (1) we use *DTDtoLDG()* algorithm to transform a DTD to a LDG; (2) according to user's purpose, the LDG is optimized based on the data-centric or document-centric rules; (3) we use *LDGtoRDB()* algorithm to map the LDG to relational schema (figure 1).

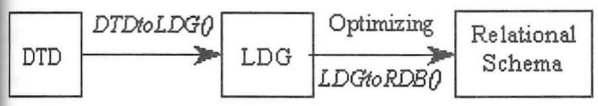


Figure 1 Overview of our approach

STRUCTURE AND SEMANTIC CONSTRAINTS IN DTD

Conceptually, a DTD is very similar to schema in relational database. When storing a documents collection to a relational database, the given DTD is mapped to corresponding relational schema, while XML documents in the collection are transformed to tuples in relational database (figure 2). In this paper, constraints we found can be classified as follows.

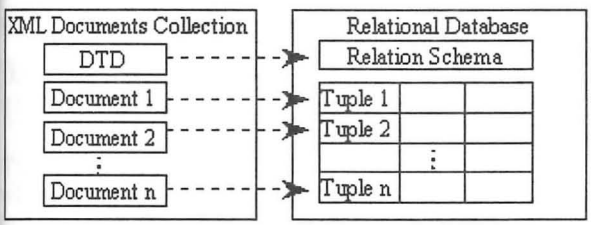


Figure 2 XML documents collection vs. relational database

Domain Constraint means that the domain of the attributes is restricted to a certain specified set of values. For instance, <!ATTLIST author gender (male | female) >.

Content Constraint is the relationship of sub-elements. For instance, <!ELEMENT A (B, C|D)> means ($C = null \wedge D \neq null$) $\vee$ ($C \neq null \wedge D = null$).

Cardinality Constraint is the relationship between an element and its sub-elements or attributes. For instance: <!ELEMENT conf (title,date,editor?,paper*) > "title" denotes one and only one "title"(1), and "paper* " denotes zero or one "paper"(0, N).

Inclusion constraint assures that values in the columns of one fragment must also appear as values in the columns of other fragments and is a generalization of the notion of referential integrity.

LDG MODEL

The LDG can be specified by the following BNF syntax:

LDG::=<NODE, RELATIONSHIP>

NODE::={n_i | n_i $\in$ Elements or Attributes in a DTD, which includes information: id, name, datatype, etc.}

RELATIONSHIP::=<CHILD-RELATIONSHIP, CARDINALITY, CONTENT-CONSTRAINT>

CHILD-RELATIONSHIP::={ (n_i , n_j) | n_i , n_j $\in$ NODE $\wedge$ n_j is a child of n_i }

CARDINALITY::=<(0, 1)> | <(1)> | <(0, N)> | <(1, N)>

CONTENT-CONSTRAINT::=<AND> | <OR>

DTD TO LDG

In this section, we show the algorithm *DTDtoLDG()* to automatically transform a DTD to an LDG. The description of *DTDtoLDG( )* is omitted due to space constraint. For more detail, see our paper.

OPTIMIZING LDG

According to user's purpose, XML documents can be classified as *data-centric* documents and *document-centric* documents.

Data-centric documents are documents that use XML as a data transport. They are designed for application consumption rather than human, so the most important thing is data rather than displaying format. *Data-centric documents* are characterized by fairly regular structure, fine-grained data, and little mixed content, and the order in which sibling elements and PCDATA occurs is generally not significant. Based on this purpose, we use the following rules to optimize LDG(Figure 3 is an optimized LDG example after using data-centric optimizing rulse.):

- A leaf Node will be split into several leaf Nodes, if its indegree >1.
- A leaf Node will be marked as attribute Node, if it matches the following conditions: 1)

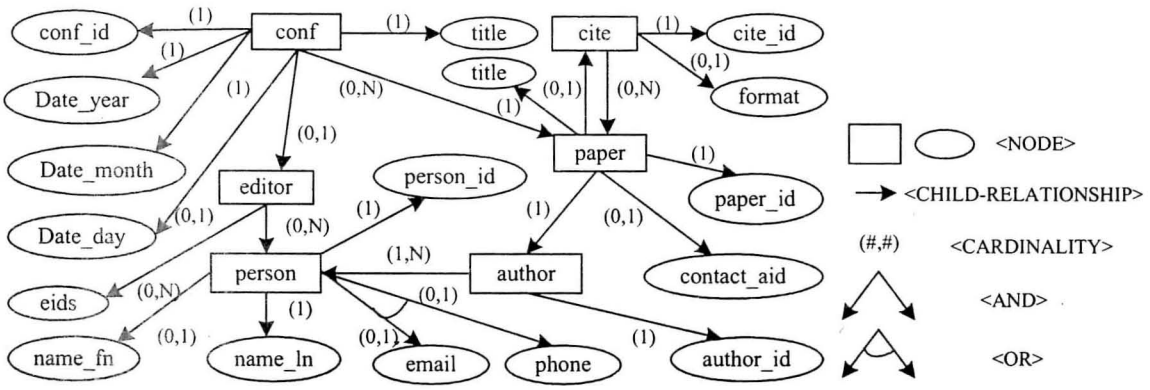


Figure 3 Optimized LDG

Node . indegree =1; and 2) Node . cardinality = (0, 1) or (1).

- A middle Node will be deleted, if it matches the following conditions: 1) Node . cardinality = (0, 1) or (1); and 2) Node . childs . cardinality = (0, 1) or (1); and 3) Node . indegree =1.

Document-centric documents are usually documents that are designed for human consumption. They are characterized by less regular or irregular structure, larger grained data and lots of mixed content. The order in which sibling elements and PCDATA occurs is almost always significant. To keep more information, we should add some information to Node, such as ordinal, depth and sibling information which will be mapped to attributes.

MAPPING LDG TO RELATIONAL DATABASE

We use *LDGtoRDB()* algorithm to automatically map LDG to relational schema. The algorithm essentially translates an LDG into SQL, and we use the CHECK, ASSERTION, TRIGGERS, etc. facility of SQL to represent semantic constraints in the LDG. The description of *LDGtoRDB( )* is omitted due to space constraint. Nodes are mapped to Table, while Attribute Nodes are mapped to attributes. Set the Node ID as the **Primary Key** of the corresponding table.

(1) Translate domain constraints as follows:
- "FIXED" and Enumeration constraints are handled with "**...CHECK (VALUE IN(...))** " clause.
- "DEFAULT" constraint is handled with "**...DEFAULT...**" clause.

(2) Cardinality constraints are handled with "**FOREIGN KEY**", "**CHECK**" or "**TRIGGER**" clause.

(3) Inclusion constraint is handled by the following method. For example, B → A: If X and Y are mapped to attributes X' and Y' in table A and B, respectively, then rewrite it as A[X']∈B[Y']. (i.e., If Y' is a primary key of B, then "**CREATE TABLE A (...FOREIGN KEY (X') REFERENCES B(Y')...)**").

(4) Non-automatically detecting constraints should be added by hand, if we need. For instance use "**FOREIGN KEY**" clause to restrict where the IDREF(S) point to, and use "**UNIQUE**" clause on unique constraint.

(5) "OR" relationship (i.e. " | " operator) is handled with "**...CHECK...**" clause. For instance, "A | B" is rewritten with "**...CHECK ((A is NOT NULL AND B is NULL) OR (A is NULL AND B is NOT NULL))...**".

We give the following SQL sentence as an example.

```
CREATE TABLE paper (
    paper_id        NUMBER          NOT NULL,
    title           VARCHAR(50) NOT NULL,
    contact_aid     VARCHAR(50),
    cite            VARCHAR(20),
    fk_cite         VARCHAR(50) CHECK (fk_cite IN
                      (SELECT cite_id FROM cite)),
    fk_conf         VARCHAR(50),
    PRIMARY KEY     (paper_id),
    FOREIGN KEY     (fk_conf) REFERENCES
                      conf(conf_id),
    FOREIGN KEY (contact_aid)   REFERENCES
                      person(person_id)
);
```

CONCLUSION

In this paper, we studied the approach of mapping DTD to relational schema both in structural and semantic aspects. Our mapping method is an improvement to the existing transformation algorithms. We focus on not only structural, but also semantic constraints. Precisely speaking, no mapping method can catch all semantics in DTDs. The purpose of our method aims at catching semantics as much as possible.

REFERENCES

1. Bray, T., Paoli, J., Sperberg-McQueen (Eds), C. M., Oct. 2000. "Extensible markup Language (XML) 1.0 (2nd Edition)". W3C Recommendation.
2. Dongwon Lee, Wesley W. Chu: CPI: Constraints-preserving Inlining Algorithm for Mapping XML DTD to Relational Schema, Journal of Data and Knowledge Engineering, August 24, 2001.
3. Deutsch, A., Fernandez, M. F. Storing Semistructured Data with STORED. In: ACM SIGMOD, Jun, 1998.
...

A Document Categorization Algorithm Using The Fuzzy Set Theory And Hierarchical Structure Of Documents

Hye-Jue Eun, Seok-Woo Han, Doo-Hyun Kim, Yong-Sung Kim, OK-Chang, Cheol-Jung Yoo
Dept. of Computer Science College of Natural Science
Chonbuk National University, Chonju 561-756 Korea
{mysiil, kdh, yskim, okjang, cjyoo}@mail.chonbuk.ac.kr, swhan@sky.wkhc.ac.kr

In present, Information retrieval systems which are simply expressed with combination between keywords and phrase search according to the direct keyword matching method to get the information which users need. But Web documents retrieval systems serve too many documents because of term ambiguity. Also it often happens that words with several meanings occur in a document, but in a rather different context from that expected by the querying person. So the user should need extra time and effort to get more close documents. To overcome these problems, in this paper we propose an information retrieval system based on the content, which connects documents according to the degree of semantic link which it express fuzzy value by fuzzy function. Also we propose an algorithm which it produce the hierarchical structure using the degree of concepts and contents among documents. As result, we are able to select and to provide user-interested documents.

Almost every document has a certain hierarchical structure concerning the importance of the words or concepts occurring in it. It can be assumed that every document has a title, which contains certainly relevant information concerning the contents. Most documents also contain sub-titles, etc. and some of them have a collection of keywords at the beginning of the text. The figure 1 is the hierarchical structure document.

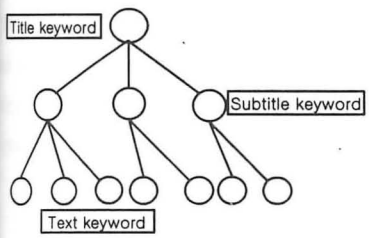


Fig. 1. Hierarchical structure of document

Hierarchical structure creation algorithm between documents

In this paragraph, we explain the hierarchical structure of the document, which it reflects the degree of mean, association, concept between documents

If the arbitrary document with the fig.1 is composing a hierarchical structure, by the sigmoid function there is a possibility of the keyword important degree according to the occurrence frequency of each location. Therefore, the value of important degree of keyword can express the relation degree. Namely, similarity degree between the document t will be able to compose a hierarchical structure.

The follow algorithms is hierarchical structure creation algorithm between the document

Step 1 : Keyword occurrence location information and frequency calculation

```
Procedure keywords_ relation( )
   doc_ id := 1;
   Index_ id := 1;
Begin
   repeat
        doc_Id : = doc_id + 1;
        Pre_processing (doc_id) ;
Extract_ keyword (doc_id);
Location (index_id)
Frequency (index_id);
Index_id : = index_id + 1;
until doc_ id : Maxdoc_id
End
```

Step 2 : Calculation relationship of Keyword occurrence location and important degree

```
Procedure degree(    )
  Num := 1, 2, 3;
   begin
Calculation_degree( )
repeat
```

```
Frequency (index_ id)
Switch(num)
Case 1 : if (index_location ==title or keywordset)
Select_ sigmoid(S1)
        Case 2 : if (index_location ==abstract or
conclusion)
Select_ sigmoid(S2)
Case 3 : if (index_location ==text)
Select_ sigmoid(S3)
Index_id := index_id +1;
Until index_id = Maxindex_id
End.
```

Step 3 : Hierarchical structure creation which uses the similarity degree between the document

```
Procedure hierarchical-structure(    )
doc_id := 1
   begin
repeat
  select_category(field)
similarity_degree( doc_id, doc_id+1)
if (degree (doc_id, doc_id +1)==1 &&
degree(doc_id, doc_id +1)== degree(doc_id+1, doc_id)
&&
degree(doc_id, doc_id+2) >=
min{degree(doc_id,      doc_id+1),      degree(doc_id,
doc_id+1) } )
          create-similarity(doc_id, doc_id+1)
insert_leaf(doc_id)
else if (degree (doc_id, doc_id +1)==1 &&
degree(doc_id, doc_id +1)== degree(doc_id+1, doc_id)
&&
             create-compatibility(doc_id, doc_id+1)
               insert_child(doc_id)
            sort(similarity-degree)
doc_id := doc_id +1;
    Until doc_id = Maxdoc_id
End.
```

The follow figure 2 shows that the documents, which they compose the brother nodes of tree structure. Also, the documents, which they satisfy compatibility relations compose a child node from tree structure.

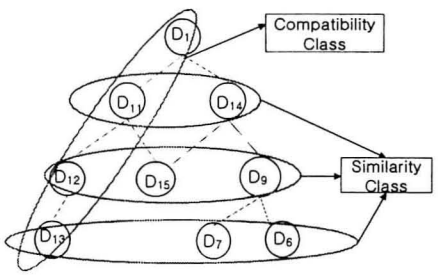


fig. 2 Construction of hierarchical structure

In this paper, we suggest query extension which user interesting is reflected to extract keyword stand for each document's content and document categorization algorithms which construction same category with document connected semantically.

Suggested query extension and document algorithm differ from previous method in view of use keyword's occurrence frequency which extracted in document and define fuzzy membership function which keyword's importance degree and define fuzzy relation between keywords and documents. Also, different membership function defined to give keyword's weight according to occurrence area extracted keyword and we try to reflect user interesting. If membership degree has satisfied fuzzy similarity relation, Query extension method constructed similarity class which consist of semantically connection between keywords that similarity degree of keyword's more than α-value in each document. Document categorization method calculate similarity degree of between documents for keyword of similarity class construed to extend query and it construct same category for semantically connected documents to classify document among higher similarity degree.

Consequently, information retrieval method applied query extension and document categorization algorithms, which they suggested in this paper offer document reflected user interesting much. Also, these algorithms offer appreciate documents more than information retrieval method of simple keyword directly matching by document, which it semantically connected classify same document set.

The further research will study automatically indexing method in constructed document set using document categorization algorithm suggested in this paper. Also we will be able to apply hierarchical classification for connected document in based on content. Consequently, we will be improve to retrieval speed, precision and recall of similarity documents using index which stand for document set use. Also we classify automatically documents to much category connected semantically much when arbitrary document stored in database.

Improving Communication Quality with Reed Solomon Code in Internet Voice Broadcasting System

Shingo Kashima* **Yukinori Goto†** **Keijiro Araki‡**
*‡Department of Computer Science and Communication Engineering,
Graduate School of Information Science and Electrical Engineering, Kyushu University
†Faculty of Economics, Kyushu University
*6-10-1 Hakozaki, Higashi-ku, Fukuoka 812-8581, Japan
*kashima@dontaku.csce.kyushu-u.ac.jp

October 24, 2002

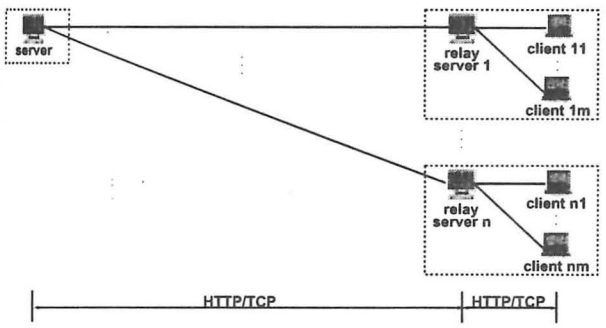


Figure 1: Existing Model

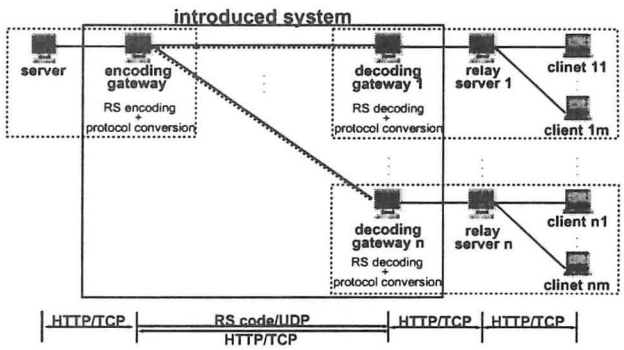


Figure 2: Implemented Model

1 Introduction

Music delivery by Icecast has diffused. Icecast delivers music to a lot of users by relay servers in real time as Figure 1. However, the connection is cut off and the music pauses because of retransmission control of TCP (Transmission Control Protocol) in bad condition of network. TCP is the transport layer protocol of communication between a server and relay servers. We have proposed replacing TCP with UDP (User Datagram Protocol) in communication between a server and relay servers and used Reed Solomon code of FEC (Forward Error Correction) to give guarantee of communication quality. We have introduced and implemented the gateways into the existing system preserving the existing applications, and finally evaluated it in real network.

2 Implementing of System

We implemented by inserting the system shown in previous section into the system of Icecast as in Figure 1. We put encoding and decoding gateways between server and relay server and implemented these gateways in order to improve the communication quality between server and relay server. The communication quality of a client is also improved without changing Icecast. The implemented model is shown in Figure 2.

Here, the protocol used for communication between each node is described. The transport layer protocol is TCP and the application layer protocol is HTTP in the communication between client and relay servers, between relay server and a decoding gateway, and between encoding gateway and server in order not to change the applications of server, relay server and client. Moreover, since there is much packet loss between encoding and decoding gateways, the data which is Reed Solomon encoded is transmitted using UDP. However, TCP is used for a transport layer protocol in order to make each message certain in the communication between encoding and decoding gateways.

The main functions of encoding gateway are performing Reed Solomon encoding to the MP3 voice data received using the connection of TCP from server and transmitting the encoded data to decoding gateway using UDP.

The main functions of decoding gateway are performing Reed Solomon decoding to the encoded MP3 voice data received using UDP from an encoding gateway and transmitting the decoded data to relay server using the connection of TCP.

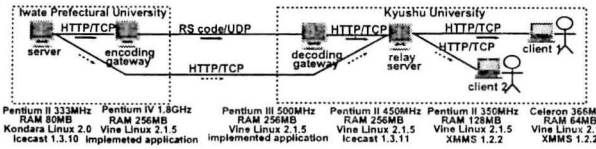


Figure 3: Evaluating Environment

3 Evaluating Experiment

This experiment compares the communication quality in the model which have Reed Solomon encoding and protocol conversion between the implemented gateways with the one in the existing model using TCP. The model using TCP is the model that server directly communicate with relay server using HTTP/TCP without encoding and decoding gateway.

Since we have to put two models in the same condition in order to compare the communication quality in two models fairly, we made environment like Figure 3 in order to communicate by two communication systems simultaneously on the same communication line.

We used MP3 data of 32Kbps in this experiment. Reed Solomon encoding in an encoding gateway is 8 bit $(32, 16)$ Reed Solomon encoding. And Moreover, payload size of the UDP packet transmitted to decoding gateway from encoding gateway is 272 bytes. Both the buffer sizes of the clinet applications in two models are for 3 seconds.

There are two measures for evaluating communication quality. The first is how many times sound pauses or noise occurs. This is performed by method of counting the number of times sound pauses or noise occurs while users hear it with their ears at client. We measured for 5 minutes. The other is connect-able time. This is the time after a client connects with relay server until disconnected.

Firstly, the graph of the number of times sound pauses or noise occurs is shown in Figure 4. A white bar shows how many times sound pauses or noise occurs in the model using TCP and a black one shows the one in the case of the model using UDP and Reed Solomon code. The number of the times sound pauses or noise occurs in the model using UDP and Reed Solomon code was from 15% to 44% for the number of times sound pauses or noise occurs in the model using TCP.

Secondly, in the experiment about connect-able time, the connection was cut off in the model using TCP in 5 minutes after the experiment starts in the worst case. On the other hand, the connection was not cut off even if the the experiment continued 150 minutes or more in the model using UDP and Reed Solomon code at this time.

Figure 4 shows that the number of times sound pauses or noise occurs decreased 72% on average, 56% at minimum, and 84% at maximum compared to the model using UDP and Reed Solomon code with the model using TCP. The experiment result of connect-able time shows that the connection in the model using UDP and Reed Solomon is harder to be cut off than the one in the model using TCP.

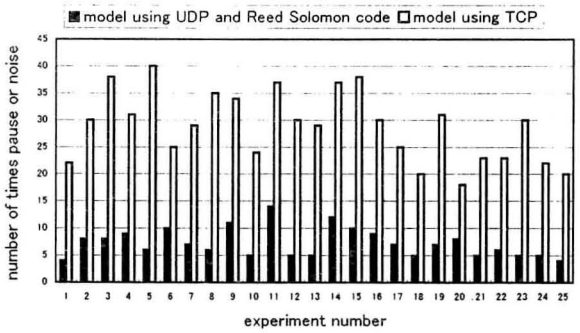


Figure 4: Comparison of the number of times of pause or noise

4 Conclusion

We aimed to improve the communication quality of the large-scale voice broadcasting on the Internet without changing the existing application. We used Icecast which uses TCP and implemented the gateways which convert the transport layer protocol from TCP, which is unsuitable for real time communication because of delay by retransmission control into UDP, which is suitable for real time communication because of no delay by the retransmission control though unreliable for packet loss, and guarantees for packet loss by using Reed Solomon code.

We evaluated the implemented gateway by using the line with much packet loss between Iwate Prefectural University and Kyushu University. In this evaluation, as a result of comparing the communication quality of the model using UDP and Reed Solomon code with the one of the model using TCP, the communication quality in the model using UDP and Reed Solomon code is better than the one in the model using TCP in the respect that pause and noise are little and connection maintains. And, the first problem "connection is cut off" among the problems in the communication quality of the Internet radio station FOR in Iwate Prefectural University was solved by using the implemented gateway. About the other problem "voice pauses and noise occurs", although some pause and noise were generated, the number of times voice pauses and noise occurs decreased, and on the whole, the communication quality of FOR was improved.

An Experience Report on Using the Team Software Process[sm]

Miguel A. Serrano
masv@cimat.mx
www.cimat.mx/~masv

Carlos Montes de Oca
moca@cimat.mx
www.cimat.mx/~moca

Centro de Investigación en Matemáticas (CIMAT)
Guanajuato, México

1

The Company

- QuarkSoft.
- Startup outsourcing software development company.
- Committed to develop quality software projects on time and within budget.
- Base its operations on PSP & TSP.

2

TSP

- TSP provides an operational process to help software engineers to do quality work.
- TSP gives the mechanisms to maintain an effective team-working environment.
- TSP provides the forms, instructions, processes, and scripts that the engineers should follow to do disciplined and effective teamwork.
- Teams working under TSP go through several main activities such as Project Launch, Running the Project, Project Relaunch, and Post Mortems (PM).

3

THE PROJECT

- Contracted by a large information and ratings service company.
- Major upgrade of a legacy system (Query Engine).
- Involves reengineering and new functionality.
- Involves databases and compiler techniques over a distributed environment.
- Development time frame of 8 to 9 months.

4

Negotiating the Project (1)

- QuarkSoft had to make an estimate and a quote.
- Factors such as price, delivery time, features, and quality were taken into account.
- TSP techniques were used to make the initial estimate and quote.
- TSP strategy requires a good conceptual design along with good historical data.

5

Negotiating the Project (2)

- Determine effort from the size estimates of product and components.
- Quality Plan Proposal sets the yield target for each life cycle phase, in terms of defects/SLOC.
- Yield assists in estimating the amount of defects that would be in the product during integration test and system test.
- Effort estimate includes the estimated amount of testing time according to yields in development phases.

6

Negotiating the Project (Final Outcome)

- Reduce 30% of functionality in order to meet the deadline and budget restrictions.
- The Project would be a 9-month 4-engineer project.
- Project would be developed at QuarkSoft´s site (controlled environment).
- Assembled Team (Maximus): PSP training (except one), TSP experience (only one w/1 yr. experience).

7

Launch

- Typical TSP Launch: 9 meetings (3-4 days).
- M2: Only five of the eight roles that TSP defines were assigned because the first cycle was planned to write the Software Requirements Specification (SRS) and the Statement of Work.
- M3: Implement The Project in 7 cycles.
  - In cycle 2 a prototype would be developed to determine the technology required by The Project.
  - In cycle 3, build the High Level Design (HLD).
  - The rest of the cycles would be for detailed design, implementation, system integration, and system test.

8

Running the Project

- Maximus did estimation errors up to 300% during the early cycles due to the lack of historical data from similar projects.
- The Project was quite different to everything Maximus' team members had done before. The Project included new technology. Maximus did a poor estimation on the technology learning curve.
- During cycles 2 and 3 Maximus under estimated by more than 50%.
- By the end of the second month, according to the Earn Value prediction, the project was behind schedule by four weeks.
- Reorganization of team due to poor performance.

9

Lessons Learned (1)

- Using the TSP approach to project estimation helped to have a more realistic estimation of time and effort. It gave solid arguments for negotiations.
- Necessary a good conceptual design and historical data on similar projects.
- "...from the beginning, all team members know the activities that each one would perform, the sequence and dependencies of them and the time they would take." --Team Leader
- "Relaunches are a fundamental part of TSP. They allow us to do detailed planning for short periods, as opposed to do detailed planning for a whole nine-month project".

10

Lessons Learned (2)

- Information available at the right time, for decision making.
- Metrics and indicators are helpful for early detection of problems and to adjust planning.
- In practice, Maximus did not have the time to perform all the data analysis that the team wanted.
- Need to develop a tool to help them do the analysis besides the conventional prototype TSP tool.
- Most of the decisions are made by team members.
- All team members and other stakeholders know the status of the project.

11

Conclusions and Recommendations

- Lack of understanding on TSP estimation and planning details might pick on productivity rate.
- If lacking historical data, it is convenient to add a percentage to account for the estimation error.
- Only thing that mattered to the client was to get the project on time.
- Necessary to have a good TSP coaching since the beginning (to avoid reinventing the wheel).

12

Modeling of Software Configuration Management Processes by using Petri nets

K. Mahesh Kumar, Dr. Gargi Keeni
SDF building salt lake electronics complex, Sec-V, Kolkata 700 091, India
mahesh_kuruba@mumbai.tcs.co.in, gkeeni@mumbai.tcs.co.in

ABSTRACT
Software Configuration Management (SCM) is of great concern in the software engineering community due to its importance in the quality and productivity of the software development. This paper presents an application of petri-net in Software configuration management. Petri-net is an established tool for modeling and analyzing systems. Petri-nets are being used for software requirements analysis and design. However the application of the petri-nets to the project management process areas is limited. Hence, this paper attempts to model the software configuration management process using petri-net. Petri net have been applied to the SCM models from literature. The missing links in the SCM models have been identified and the modified petri nets has been presented. A model of this kind is very essential for the SCM tool developers as well for the organizations deploying SCM tool. Organizations deploying such SCM tools can map their existing process to the tool or follow the process of SCM tool.

KEYWORDS
Software configuration management, Petri nets, Change management model

BRIEF SUMMARY OF PAPER
Petri net has been applied to three SCM models, which are identified from the literature. The missing links in the model has been identified and the necessary changes in the model have been incorporated in the petri net to make it complete.

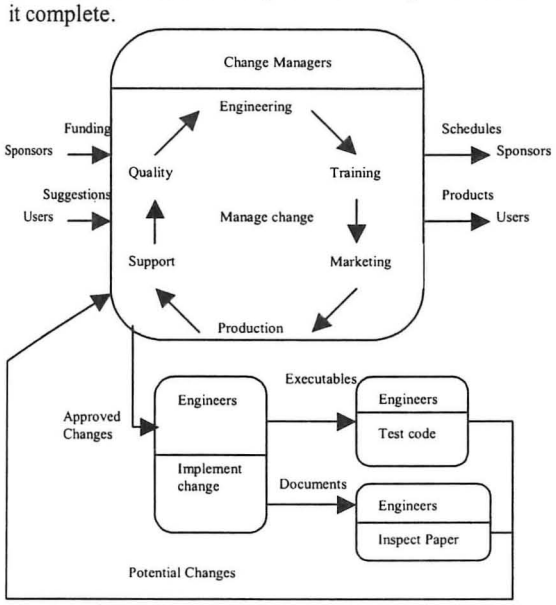


Fig.1. Oleson's change management model

The three SCM models are:
1. Olsen's change management model
2. V-like change management model
3. Ince's change process model

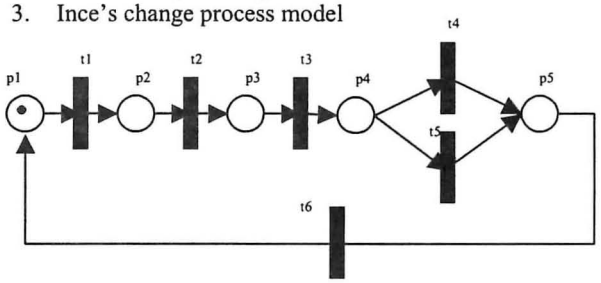


Fig. 2. Petri net of Olsen's Model

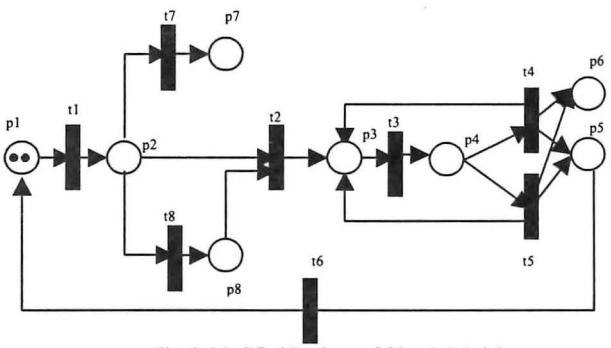


Fig. 3. Modified Petri net of Olsen's Model

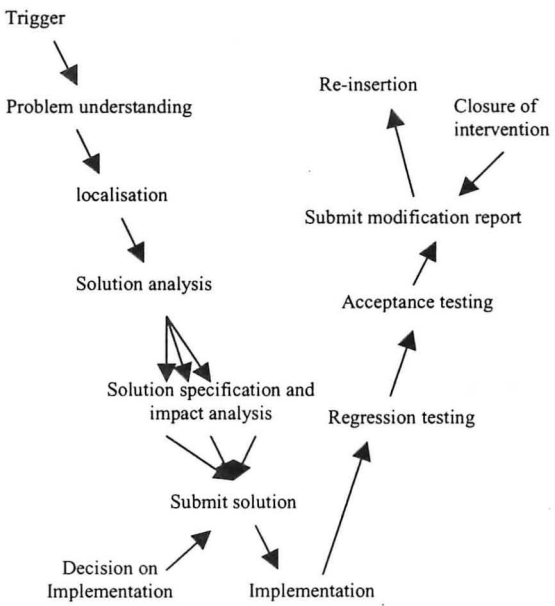


Fig.4. V-like change process

The following missing links have been observed in the Olsen's model:

- What will happen to the deferred or rejected changes?
- How will the manager measure the no of change requests completed?
- What if defects are uncovered during the execution of code or inspecting the documents?

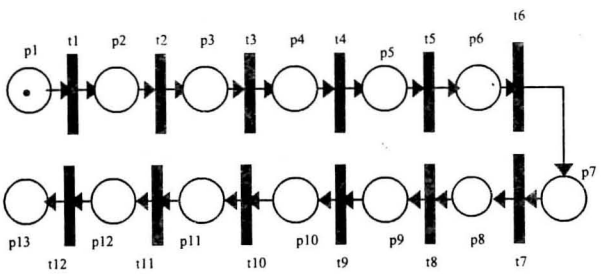


Fig.5. Petri net of V-like change process

From fig.5, it can be observed that the following links are missing in the V-like CM model:

- The decision of not implementing the solution
- Defects uncovered during regression or acceptance testing
- Defects injected due to the change

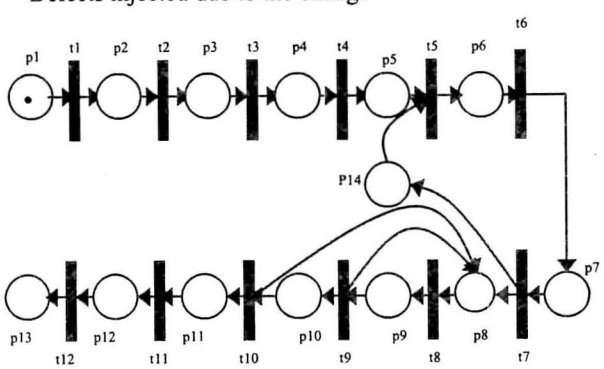


Fig.6. Modified petri net of V-like change process

From fig.8, it can be observed that the following links are missing in the Ince's model:

- Deferred changes
- Defects uncovered during validation of CR

Modeling SCM processes using petri nets would be helpful to simulate and identify the missing links in the model. The ability of extending the petri nets to stochastic petri nets, enhances its capability to analyse and simulate the SCM processes. Also, it enhances the understanding of the SCM tool developers about the SCM processes.

CONCLUSION:

This paper proposes a petri net based approach for software managers, in evaluating and selecting the appropriate SCM processes or SCM tools for their organization or project. Three SCM models from the literature has been selected and petri net approach has been applied for the models to identify the missing links in them. The identified missing links has been incorporated and the modified petri nets has been illustrated. This paper also proposes to use this kind of approach to develop SCM tools.

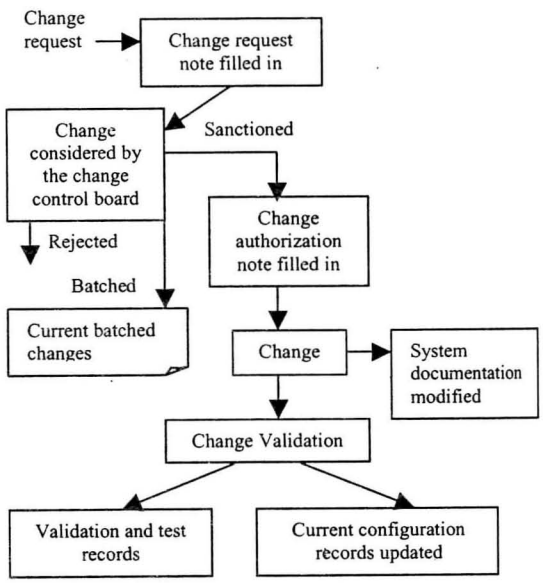


Fig. 7.Ince's change process

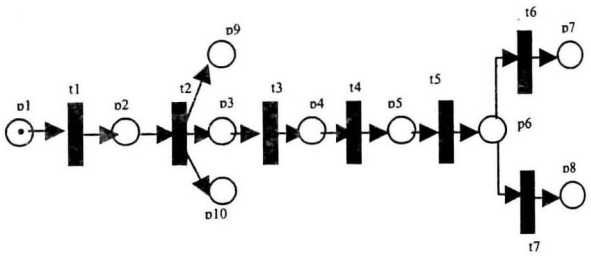


Fig.8. Petri net of Ince model

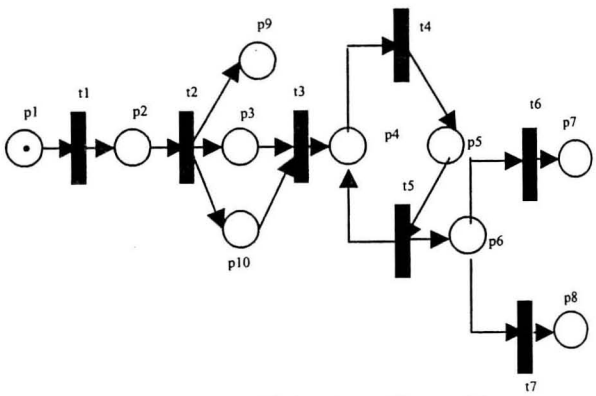


Fig. 9 Modified petri net of Ince model

On the Effectiveness of Genetic Operators in Adaptive Random Testing

S.P. Ng*, F.C. Kuo, R.G. Merkel, T.Y. Chen
School of Information Technology,
Swinburne University of Technology, Hawthorn, Victoria 3122, AUSTRALIA

Email: {sng, diana_kuo, rmerkel, tychen}@it.swin.edu.au
Phone: +61-3-9214-8666, Fax: +61-3-9818-0823

Abstract

Adaptive random testing (ART) has recently been found to be an effective way to improve the performance of random testing by selecting *widespread* test cases from the input domain. In this paper, we outline how the selection, mutation and crossover operators of the genetic algorithm (GA) can be used in ART, and investigate experimentally which of these is the most appropriate operation to serve for the purpose of widespread as required in performing ART.

Paper Summary

Adaptive random testing (ART) is a recently developed method to improve random testing, for the common cases of contiguous failure-causing regions.

A basic version of ART can be described as follows. Suppose $T = \{T_1, T_2, \ldots T_m\}$ is the set of executed test cases. Then, randomly generate a set of candidate test cases $C = \{C_1, C_2, \ldots C_n\}$ from the input domain. Determine the candidate C_k of C, such that C_k is *furthest* away from the elements of T, as the next test case to be executed. If C_k does not reveal any failure, add it into the set T and the process is repeated until a failure is detected or testing terminates according to some pre-defined criteria, whichever earlier.

Genetic algorithms have been shown to be a well-established iterative approximation method for finding near-optimal solutions to some hard-to-solve problems. In this paper, we examine methods for performing the three basic operations of GA (crossover, mutation, selection) in the context of ART. We also present an empirical analysis of the effectiveness of these basic operations.

In this investigation, we only considered a program whose inputs are integers. For simplicity, we further assumed that the input domain consisted of two variables, both restricted to the range [0, 127]. Hence, a test case can simply be represented as a 14-bit vector.

In this paper, we describe a method of applying the basic genetic operators in ART, which we name Genetic-ART (G-ART). In this method, two sets of program inputs are recorded - the set of inputs already used to test the program (known as the *executed* set), and a set of as-yet unexecuted potential test inputs (known as the *candidate* set).

Members of the candidate set are evaluated by a fitness function in light of the executed set and the candidate with the highest fitness is chosen. A test is then carried out using the chosen candidate. If the test detects a failure, the procedure halts. Otherwise, the input is added to the existing executed set and a new set of candidates is then generated by applying the basic genetic operators (selection, mutation, and crossover) to the current candidate set, and the process iterates until failure is detected or maximum predetermined number of tests are performed.

The fitness function used was based on the Cartesian distance between each candidate test case and its closest already-executed test case:

$$fitness(c_j, E) = \min_{i=1,\ldots m} \{dist(c_j, e_i)\}$$

where E is the executed set of test cases, e_i is an element of E, c_j is an element of the candidate set C, m is the size of E, and $dist(p,q)$ denotes the Cartesian distance between two points p and q in the input domain.

* Contact Author

Mutation is a process by which new candidates are generated by somehow modifying an existing candidate. In our simulations, mutation was performed by simply flipping two randomly-chosen bits of the 14-bit vector. Crossover is a process by which new candidates are generated by combining two old ones. In our proposed crossover method, the 14-bit vector of the new candidate produced by two candidates randomly picked from the original pool was made up of three portions: the first and the last portion came from one of the two candidates and the middle portion came from the other. In the generation process, the first or the last portion may be empty. Selection is a process by which a new candidate is generated by simply selecting an old one. In our experiments, selection was performed by choosing those in the old candidate set with the highest fitness.

A simulation study was conducted to determine the effects of the basic genetic operators on the failure-detection capabilities of the G-ART algorithm.

Our study examined G-ART in the case of a 2-dimensional integer domain of size 128 x 128 as previously described, with a randomly chosen square failure-causing region of size 4 x 4. The number of candidates was in all cases fixed at 10.

To study their relative influences, the proportion of the candidates generated by selection, mutation and crossover was varied systematically in our simulation.

The number of experimental trials required for each condition was determined by calculating the confidence interval for the mean F-measure. In our simulations, we have chosen a confidence level of 95% and accuracy range of ± 5%.

Table 1 shows the effect on the expected F-measure by varying the candidate proportion generated by selection and crossover, with those by mutation fixed at 20%, 40% and 60%:

Mutated	Selected	Crossover	Trials	Expect(F)	Std Error	95% Confidence Int.	
						-	+
20%	20%	60%	1080	703	17.8	668	738
20%	40%	40%	1120	748	19.1	711	785
20%	60%	20%	1220	751	19.1	713	788
40%	10%	50%	980	650	16.5	618	683
40%	30%	30%	900	684	17.3	650	718
40%	50%	10%	960	706	17.9	670	741
60%	10%	30%	980	639	16.2	607	671
60%	20%	20%	980	626	15.9	595	657
60%	30%	10%	1000	675	17.2	642	709

Table 1

From the results shown in Table 1, it can be observed that if the proportion of mutated candidates is fixed, altering the proportion of candidates generated by crossover and selection has no significant effect on the expected F-measure. However, there is an observable trend that increasing the mutation proportion tends to reduce the expected F-measure.

Table 2 shows the effect on the expected F-measure by varying candidate proportion generated by mutation and crossover, with those by selection fixed at 20%, 40% and 60%:

Mutated	**Selected**	Crossover	Trials	Expect(F)	Std Error	95% Confidence Int.	
						-	+
60%	**20%**	20%	980	626	15.9	595	657
40%	**20%**	40%	920	670	17.0	637	704
20%	**20%**	60%	980	703	18.7	666	740
50%	**40%**	10%	960	663	16.8	630	696
30%	**40%**	30%	980	676	17.0	642	709
10%	**40%**	50%	1280	907	23.1	862	953
30%	**60%**	10%	1060	733	18.6	697	770
20%	**60%**	20%	1220	751	19.1	713	788
10%	**60%**	30%	1440	965	24.5	917	1013

Table 2

The results in Table 2 show that, for fixed proportion of selected candidate, increasing the proportion of candidate generated by mutation and decreasing that generated by crossover reduced the expected F-measure. There is also an overall trend that the smaller the proportion of candidates generated by selection, the smaller the expected F-measure.

Table 3 shows the effect on the expected F-measure by varying candidate proportion generated by mutation and selection, with those generated by crossover fixed at 20%, 40% and 60%:

Mutated	Selected	**Crossover**	Trials	Expect(F)	Std Error	95% Confidence Int.	
						-	+
60%	20%	**20%**	980	626	15.9	595	657
40%	40%	**20%**	1020	686	17.4	652	720
20%	60%	**20%**	1120	751	19.9	712	790
50%	10%	**40%**	960	655	16.5	623	688
30%	30%	**40%**	1020	673	17.0	639	706
10%	50%	**40%**	1240	965	24.6	917	1014
30%	10%	**60%**	1100	696	17.2	662	730
20%	20%	**60%**	1080	703	17.8	668	738
10%	30%	**60%**	1280	920	23.5	874	966

Table 3

The results in Table 3 show that if the proportion of candidate by crossover is fixed, increasing the proportion of candidates generated by mutation and decreasing that by selection reduced the expected F-measure. In addition, decreasing proportion of candidates generated by crossover tends to slightly reduce the expected F-measure, but not to the extent produced by the selection operator as observed in Table 2.

Research of New Web Application Skeleton Based on XML/CORBA

Fu Xiufen [1,2], Wang Jing [2], Lu Yansheng [1], Sun Jianfang [2], Yang Junchao [2]

1 Huazhong University of Science & Technology,Wuhan, 430074, P.R.China

2 Guangdong University of Technology, Guangzhou, 510090, P.R. China

ABSTRACT

XML is the hot spot in research of Web e-business applications, CORBA can solve problems of inter-platform, which implements the distributed software integration. The paper discusses compatibility problems of XML used on Web and resolvents based on CORBA, designs an application system based on the new XML/CORBA Web skeleton–XML Wrapper system, discusses the designing goal, system structure, key technologies and implementations, shows the characteristics of XML Wrapper system in the end.

1. INTRODUCTION

XML Wrapper is an XML/CORBA-based Web application system developed by us. According to the design and implementation of XML Wrapper, this paper discusses problems of XML-based Web applications and gives our CORBA-based solution. This paper also illustrates that Web applications using XML/CORBA is easy to develop and easy to use. It will impel the research of XML/CORBA-based Web applications.

2. APPLICATION OF XML ON WEB

XML is an optimized subset of SGML. Present Web applications adopt a three-tier structure.

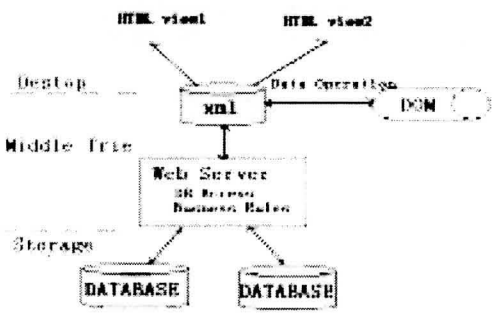


Fig.1 XML data exchange structure

The main function of storage-tier is the management of data storage, through the background database system. Data can be made up of asynchronous data sources. The middle-tier can access data and turn them into XML data through interfaces provided by storage-tier, implement the business rule and deal with XML data at the same time. The display-tier can control the foreground display of XML data according to predefined XSL files.

3 COMBINATION OF CORBA AND WWW

CORBA is a standard architecture defined by OMG to realize objects with different languages and platforms. CORBA provides excellent characteristics of inter-platform, inter-language and location transparence, shields complex distributed asynchronous environment, realizes integration of distributed software wholly, also provides a lot of usable distributed services, so greatly simplified implementation of the system. The integration skeleton provided by CORBA brings systems based on it advantages of clear structure, easy to manage and maintain, so that the inter-operation of objects in distributed asynchronous environment is ensured. CORBA is the best choice to combine distributed object technology and WWW. On the other hand, applying CORBA into XML applications can combine the great transportation ability with agility of XML well, promote the manageability, scalability, maintenance and integration ability of the system greatly so provides greater functions to WWW applications.

4. XML WRAPPER SYSTERM DESIGN

XML Wrapper is a new application based on XML/CORBA new Web skeleton. The system is based on CORBA platform, integrates data of different sources into uniform XML data, inserts an XML layer between users and data sources and responds users by parsing XML documents. XML Wrapper shows well the advantage of XML that content and form are separated and the component idea of CORBA.

4.1 Designing Goal

The goal of XML Wrapper system: turn data from different databases into XML documents; store, query and update XML documents; store the change of XML documents to databases; manage DTD and explore Schema; make the system inter-platform based on CORBA.

4.2 Structure Design

XML Wrapper is a new Web application based on XML/CORBA. Based on the "soft bus, soft component" skeleton of CORBA, the structure of XML Wrapper system is shown in Fig. 3.

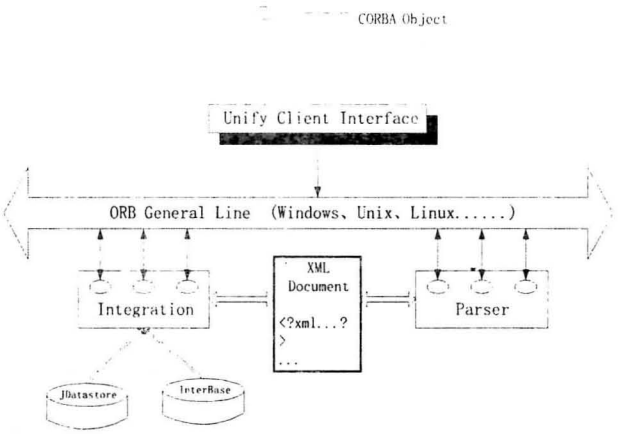


Fig. 3 system structure

The system is made up of six parts including distributed databases, XML integrator, XML parser, DTD module, uniform user interface and object request broker ORB.

The user interface submits query as a CORBA user after it receives users'input; XML integrator integrates data from related distributed databases into uniform XML documents according to some DTD; XML parser parses the document to respond users and send the results to users through user interface or to distributed databases through XML integrator. XML parser has double roles of CORBA server and user in the whole process. What users see are uniform XML documents because XML integrator has shielded the different structures of distributed databases. The design of the system adopts a new idea of data integration.

5. THE CRITICAL QUESTIONS OF THE SYSTEM AND RESOLVENTS

5. 1 XML integration of different databases

The system adopts XML as the public model of the integration system, DTD to describe virtual classes of integration layer, and elements correspond to attributes of virtual object classes. All DTD of virtual object classes form integration modes of integration systems.

5. 2 Public data format

We have designed a public data format for XML Wrapper system descripted in XML. This public data format is designed to be self-descriptive, define datatype at running time, describe relational data modes and dynamic change in horizontally and vertically.

5. 3 IDL interfaces of the system

XML Wrapper uses IDL to describe above public data formats . Datatypes used in IDL files and mapping of them to Java language include basic datatype mapping and structure datatype mapping.

5. 4 Parsing of XML documents

The XML Wrapper system mainly adopts DOM method .On the other hand the system has some innovation based on this. For example SAX is used in query operation. The system parser is JAXP of Sun.

5. 5 The display of XML

The system has used XSL as the XML document display language.

5. 6 IDL objectifying of XML documents

We have designed an encapsulation policy based on tables of databases to encapsulate XML documents into CORBA objects. The object names and encapsulated XML documents are corresponding one to one and objects are named by XML document names. From the view of general environment we only can see objectified sharing data documents one by one, shown in figure 4.

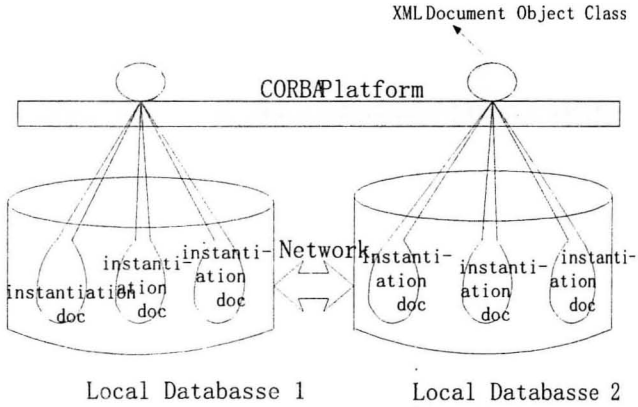


Fig.4 encapsulation of XML Document

5.7 CORBA naming service

In XML Wrapper system the naming service is used to locate objects and functions are used to implement the transformation of object references and names.

6. SYSTEM CHARACTERISTICS

- **Expansibility, opening and reusable**
- **Adaptability**
- **Plug and play**
- **Maintenantability and managablity**
- **Easy to implement**

7. CONCLUSIONS

The paper has combined the two technologies to construct a new web application framework—XML Wrapper system. It has separated database, application logic and client effectively and has formed a good three-tier server architecture so that network applications are maintained and expanded more easily, load on server side is greatly released to give attention to server and client and is facilitate to data platform independence. Importing CORBA as the middleware to the XML applications and using CORBA to transport XML documents improve the expansibility and maintainability of interfaces and make the display of data structure clear and easy to operate.

Organise Class Libraries based on Functionality

Huilin Ye

School of Electrical Engineering and Computer Science, University of Newcastle, Callaghan, Australia 2308

Hanchang Liu

Graduate College of Management Southern Cross University Lismore, Australia 2480

ABSTRACT

There is a very well defined need for retrieving available classes from a class library. Many reusable classes are now reinvented as the reinvention may be fast than searching. Although hierarchies of these classes are generally available, the information about how these classes are functionally related is difficult to access. However, software developers are usually more interested in the functionality of the reusable classes. This paper presents a novel method for building a software library from which the desired components can be found in terms of functionality. The proposed approach was applied to a collection containing all the classes in Java.awt package. The major advantage of this approach is that an effective library can be constructed automatically without much human intervention, i.e. it achieved an optimal balance between efficiency and effectiveness.

SUMMARY

Object-oriented programming languages provide large collections of reusable classes for the software developers. Although hierarchies of these classes are generally available, the information about how these classes are functionally related is difficult to access. However, software developers are usually more interested in the functionality of the reusable classes. It is crucial for the reuse purpose how to find in the library those classes that have the functionality required by a particular application under construction. In addition to the hierarchical relationships among the classes that are already presented in object-oriented libraries, the libraries should also be able to provide a classification and retrieval mechanism based on class functionality.

A number of techniques for software classification and retrieval have been proposed. The evaluation of these techniques is generally based on two criteria: effectiveness and efficiency. An efficient library system should be able to build and to update automatically without much human intervention. Manual approaches to building a library are difficult to scale up as considerable human effort is needed. As the demand for software keeps increasing and new object-oriented programming languages, such as Java, provide the user with a huge number of reusable modules, a small-scale software library is not capable of facilitating reuse in such an environment.

The effectiveness of a software library system can be assessed by its retrieval performance that is usually measured by *recall* and *precision*. Recall is the ratio of the number of the relevant components retrieved for a given query over the total number of components relevant to that query in a component collection. Precision is the ratio of the number of relevant components retrieved over the total number of components retrieved. Together, recall and precision measure how successfully a system retrieves all the relevant components while at the same time excluding irrelevant ones for a given query.

Unfortunately, efficiency and effectiveness are conflicting factors in a library system. Improvement on one factor may deteriorate the other. Currently existing software library systems are either too ineffective to be useful or too inefficient to be practicable.

This paper presents a new approach to building software libraries. Software documentation is used as a surrogate of software components in this approach. For the reuse purpose, the most significant information about a software component is its functionality that is usually described in an associated document. To achieve an optimal balance between effectiveness and efficiency this approach combines the strengths of the automatic free-text indexing method and the Self-Organising Map (SOM) neural network technology. The automatic free-text indexing method makes the proposed approach more cost-effective than the controlled vocabularies and knowledge-based systems, since the amount of human effort involved is significantly reduced. On the other hand, the SOM is capable of exhibiting the semantic relationships among software components while the general free-text indexing systems cannot.

The SOM is an unsupervised neural network and has an internal learning rule that enables it to learn from the input data without any supervision. After enough input data is presented, the reference vectors will specify clusters that mimic the input vectors in such a way that the point density function of the reference vector centres approximates the probability density function of the input space. No training data is needed in the learning process. Instead, the SOM learns from input stimuli. Input vectors with similar features will be mapped onto adjacent areas of the grid. It is this property of the SOM that makes it useful for classification.

According to the basic concepts of the SOM, a software library containing all the classes of Java.awt package was constructed. The library consists of a component map (Figure 1) and a semantic map (Figure 3) that can be derived from the component map. Clusters can be identified from the component map while the semantic

map is helpful for understanding the functionality of the components.

Based on the method, class libraries can be organised in terms of functionality, which is good for reuse purpose. The major advantage of this approach is that an effective library can be constructed automatically without much human intervention.

The proposed method has been evaluated and the result from the evaluation was very promising. The next step is to apply the method to a large class library and incorporate the information about hierarchical relationships among classes into the library to further enhance the usability of the library.

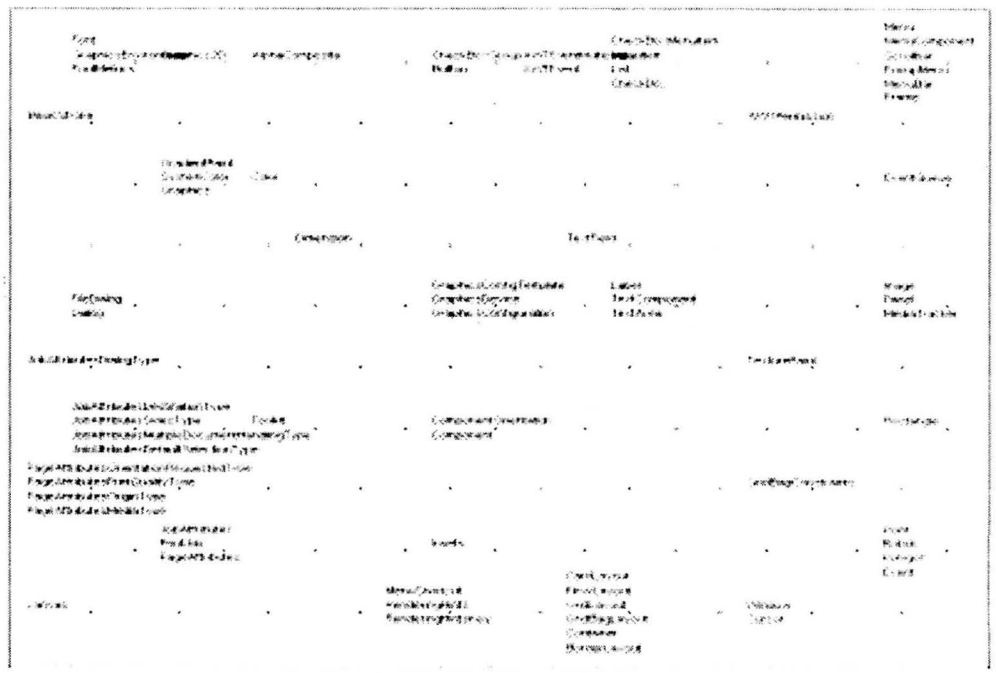


Figure 1. Component map of JAVA.AWT package

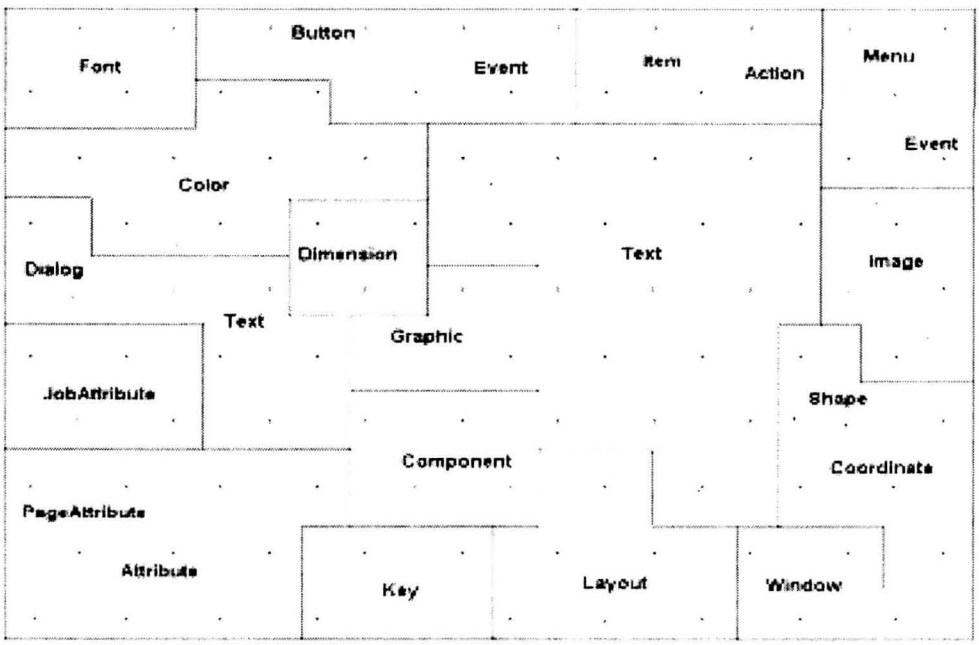


Figure 3. Semantic map of the JAVA component map

UML-based Component Design Technique

Eun Sook Cho[1]
College of Computer & Information Science
Dongduk Women's University
Seoul, Korea
escho@dongduk.ac.kr

Soo Dong Kim[2], **Sung Yul Rhew**
Dept. of Computer Science
Soongsil University
Seoul, Korea 156-743
{sdkim,syrhew}@comp.ssu.ac.kr

Chul Hong Kim[3]
Dept. of Software Engineering
ETRI
Daejon, Korea 305-350
kch@etri.re.kr

Abstract

Although component implementation technology is introduced, modeling techniques for building software components have not been studied enough. Existing methodologies don't describe concrete modeling guidelines for component analysis and design.

This paper proposes a practical component design technique for component development. Proposed technique is a part of COMO[12] method. This paper only describes modeling techniques of preliminary design phase. Especially modeling techniques related with component interface definition, component's internal design, and component's customizability design are provided in this paper. Since proposed modeling techniques provides systematic process and comprehend instruction for each development task, we believe it can be effectively used in modeling and implementing software components.

Keywords: *Component, Customizability, Component Design*

1. Introduction

Plugging reusable components can develop a component-based software. To develop such reusable components, it is desirable to have a systematic component development methodology. However, existing major methodologies do not provide concrete guidelines for component development. Therefore, it is difficult to adapt these methodologies practically into component development or component-based software development. Recently, much concern of component focuses on analysis and design method, but current framework methods only provide outline of component analysis and design. They do not provide stepwise development process and concrete analysis and design guidelines. Therefore, we develop a component development methodology based on the UML. In this paper, we only focus the design techniques for component development. Other techniques are proposed on the Ph.D thesis[12].

In this paper, we propose a practical OO component design technique that extends UML [1,2] notations and semantics. The proposed technique consists of six tasks. In order to help the seamless migration from one task to subsequent tasks, we provide instructions for carrying out tasks. The proposed process is based on UML diagrams and the basic Unified process.

The rest of the paper is organized as follows. Section 2 deals with the types of existing CBD methodologies and the limitations of those. Section 3 presents an overview of the component methodology that we developed. Section 4 describes modeling techniques of preliminary design phase in the methodology. Section 5 deals with the case study of proposed approach and the main advantages of this approach by comparing with other approaches. Section 6 gives the concluding remarks.

2. Preliminary Component Design

Preliminary design phase consists of six tasks, and these tasks are processed iteratively (See Figure 1).

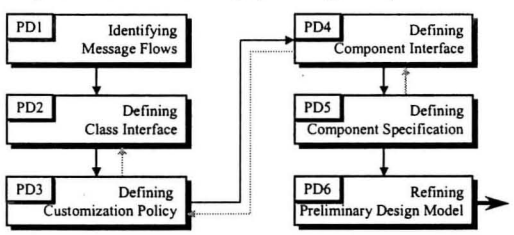


Figure 1 Preliminary Design Phase

Identifying Message Flows (PD1)

In this paper, we extend sequence diagram. Sequence diagram of UML only represents message flows between objects in a use case. However, there is message flows

between objects as well as message flows between object and component in a component. Therefore, we add component instance in sequence diagram, and represent message flows between object and component (See ***Error! Ref.***). Furthermore, message flows of sequence diagram are corresponded to the equivalent interface of CCM[9].

These message flows are extracted from use case description for each component. Each event is mapped into message flow between objects in a component.

The name of message flows for component interface is described as ' component name::interface name'.

Defining Class Interface (PD2)

Elaborating conceptual class diagram means adding operations in classes included in conceptual class diagram and adding additional objects related with component implementation. Also, detailed information such as navigability, multiplicity, constraint, or qualified association is represented.

In order to add operations in classes, we refer object sequence diagram resulted in previous task. If one object receives message from the other objects, we should map the message into methods of the object (See ***Error! Ref.***).

Defining Customization Policy (PD3)

We classify defining customization policy into three customization policies; customization policy for variant attributes, customization policy for variant logics, and customization policy for variant workflows.

Defining customization policy for variant attributes is classified into two types; one is determining customization policy for variant values of attributes, and the other is determining customization policy for variant attributes set having fixed operations. In order to determine customization policy for variant values of attributes, we should refer to as following: commonality and variability table and detailed class diagram. And then, we define methods of variant attributes as *set<AttributeName>* pattern. Otherwise, in order to determine customization policy for variant attributes set having fixed operations, we define variant attributes set as template class. In this case, any object type can be parameterized.

Defining Component Interface (PD4)

Provided interface of a component is an interface which a component provides other components or clients. Required interface of a component is an interface which a component needs.

In order to define provided interface, we determine component interface per each class contained in a component. As shown in ***Error! Ref.***, if there are message flows sent from the controller to classes, we define component interface per each class, and assign message flows into the component interface.

In order to define required interface, we select sending message flows from the controller class to object of other component or select sending message flows from the object in a component to object of other component. That is, if there are message flows between object and other component or message flows between controller and other component in a sequence diagram, we define such interface of other component as a required interface of target component.

Defining Component Specification (PD5)

We propose component contract as extended IDL of CORBA component model because IDL of CORBA is independent on specific programming languages (See ***Error! Ref.***). In IDL form, component is described by "Component" syntax, and several interfaces are contained in a component. Also, interfaces of component are classified into provided interfaces and required interfaces. Provided interfaces are described by "Provides" syntax, and required interfaces are described by "Uses" syntax. Especially, when we describe required interfaces, we add component name through scope operator. Additionally, pre-condition, post-condition, and exception are described for each operation in an interface of component.

In pre-condition section, we describe contents satisfied before the component user calls operation of component interface. In post-condition section, we describe contents satisfied after the component user calls operation of component interface. In exception section, we describe exceptions into which are occurred when the component user uses operation of component interface incorrectly. These exceptions are identified from exception flows in a use case description.

After defining component contracts, we make out component specification. Component specification consists of component name, brief description, participants, static model, dynamic model, and component contract sections.

In the participants section, classes name contained in a component is described. Component diagram is represented in the static model section, sequence diagrams are described in a dynamic model section, and component contract is illustrated in the component contract section.

3. Concluding Remarks

This research has been and still is a challenge. A number of methodological issues regarding component development have been surveyed and presented. We proposed new practical component design technique that extends UML and Unified process. Because proposed technique contains systematic design process, sequences of tasks flows, inputs and outputs of each task, validation of each phase or task, component developers can apply this design technique into practical project. Furthermore, we proposed detailed component design guidelines according to component platforms, which are used to implement components.

A Data Broadcast Scheduling Strategy for Real-time Read-only Transactions in A Mobile Environment

LIU Yunsheng YANG Jincai LI Guohui
School of Computer Science and technology, Huazhong University of Science and technology.
Wuhan 430074, P.R.China

ABSTRACT

Data Broadcasting is an important data dissemination approach in mobile environment. On broadcast channel, Scalability and Efficiency of transmission are met. In a mobile environment, there exist a kind of real-time database applications in which both the transactions and data can have their timing constraints. Many broadcast scheduling approaches have been proposed to schedule individual data item. They may not work well when the transaction may access more than one data item. In this paper, we propose a broadcast scheduling strategy for real-time transaction based on the BoD (Broadcast on Demand) model. In the strategy, the issues of both accessing multiple data items and meeting the timing constraint of the transaction are considered. The purpose of the algorithms is to reduce the number of transactions missing their deadlines.

RELATED WORK

Because of the mobility of the user, the number of users in a cell is not limited. Scalability of communication bandwidth is required. Data broadcast allows users to retrieve data simultaneously with a cost independent of user numbers. Thus, broadcast technology is widely used in mobile environment as a way of information dissemination[1]. Researches on data broadcast in mobile computing systems have received much attention in recent years. A recent study presents a data broadcasting model and associated scheduling algorithm based on the deadlines of requests. In[3], BoD model and associated scheduling algorithm such as EDF and EDF-BATCH are proposed to satisfy the temporal constraints of the request. In EDF startegy, The scheduler schedules the requests with earliest deadline first and rejects those requests that are not schedulable. In EDF-BATCH stragegy, For each candidate request, it checks to see if transmission has already been planned for the requested data. If so, it does not re-send the same data, as the planned transmission will also be able to satisfy the current request.

From the above stragegies, it is obvious that if the ultimate deadline of a transaction is used as the basis for assigning data deadline, the transaction with fewer requests will be favored. In [4] Kam-Yiu Lam et al. proposes RP (request Proportion) and EQSD (Equal slack Data) schedule algorithms which apply for disseminating data items to transactions with multiple data request. In order to allow the longer transactions to have a reasonable chance of being completed within their deadline, the deadline of each data item of a transaction is assigned as following calculation.

Deadline(di)= $\frac{deadline\ (T)}{|QDS\ (T)|}$, |QDS(T)| is the number of data items in QDS(T).

OUR WORK

In this paper, we propose some new scheduling strategies. In those strategies, not only timing constrains but also accessing frequency of data items are considered. We stress on satisfying the integrality for multiple data items accessed by transaction.

EDF-T(Earliest Deadline First for Transaction)Strategy

In EDF_BATCH strategy, data items in a transaction are assigned with the deadline of their transaction. When deadline of some data items in a transaction is met but that

of others' aren't met, the deadline of the transaction is missed. In this case some data items in the transaction aren't removed from the scheduled queue. They occupy the broadcast channel. In our strategy, we use the whole data set of a transaction as scheduling unit. When deadline of a transaction isn't met, the data items of the transaction will be removed from ready for schedule queue.

HUF-T(Highest Urgency First for Transaction) strategy

In EDF-T, transaction with earlier deadline will be scheduled first. This will benefit the transactions, which accesses fewer data items. Transaction with more data items will have higher probability of miss their deadline. So we propose the HUF-T strategy, which uses urgency as schedule priority metric.

$$\text{Urgency}(T_i)= \frac{\text{Deadline}(T_i) - \text{TBD}(T_i)}{|QSD(T_i)|}$$

Using above formula, transactions which own more data items will be assigned with a high urgency.

HCF-T(Highest Contribution First) strategy

In real application, there are some data items, which accessed by many transactions. They are called "hot" data. If they are broadcasted first, many descendent transactions, which access those data items, will be satisfied. This will help improve the throughput of the transaction. Therefore, increase the number of transactions whose deadlines are met.

We use mean contribution to evaluate the hotness of data items in transaction.

$$\text{Contribution}(T_i)= \frac{\sum_{j=1}^{n} fre(d_j)}{n}$$

In the formula, n is $|QSD(T_i)|$, $fre(d_j)$ is the frequency of d_j in wait for schedule queue.

SIMULATIONS AND RESULTS

In our simulations, we assume the deadline of transaction has an uniform distribution between min_d and max_d. Accessing probability to different data items is governed by Zipfian distribution.We produce 5 groups of data items in the experiment, and the simulation is performed in 100 seconds. Each point in the graphs is the average over 5 runs.

Result shows the strategies based on transaction are better than strategies based on data item. Using EDF-T, the SR is 4% higher than that of using EDF_BATCH.

Among EDF-T, HUF-T and HCF-T, SR for EDF-T is 2% higher than that for HUF-T, because EDF-T strategy is favored for the transaction which access fewer data items. When interarrival time is lower than 2ms, SR for HCF-T is higher than that for EDF-T. This means that HCF-T strategy achieves better performance than EDF-T under the condition of data congestion. In mobile data processing system, we can integrate the tree strategies.

REFRENCES

1. Daniel Barbara., "Mobile Computing and Databases–A Survey" IEEE Transactions on Knowledge and Engineering, VOL.11, NO. 1, January/February 1999

2. Y.D.Chuan and M.H.Kim. "QEM: A Scheduling Method for wireless Broadcast Data", Proc. International Conference on Database Systems for Advanced Applications Proceedings, page(s) 135~142

3. Ping Xuan, Subhabrata Sen, Oscar Gozalez, Jesus Fernandez and Krithi Ramamritham. "Broad on Demand: Efficient and Timely Dissemination of data in Mobile Environments" Real-time Technology and Applications Symposium,1997. Proceedings., Third IEEE,1997. Page(s): 38-48

4. Kam-yiu Lam, Edward Chan and Chun-Huang Yuen. "Data Broadcast for Time-Constrained Read-Only Transaction in Mobile Computing System. Advance Issues of E-commerce and Web-Based Information System, WECWIS, 1999. International conference page(s):11-19

Some Design Issues for an Enterprise Geographic Information System

Wei Lai*, Donggang Yu* and Yongliang Zhong+
*School of Information Technology, Swinburne University of Technology
PO Box 218, Hawthorn, VIC 3122, Australia

Email: {wlai, dyu@it.swin.edu.au}

+Hypersoft International Pty Ltd, NSW 2121, Australia
Email: yzhong@hypersoft.com.au

Summary

There are lots of Geographic Information Systems (GIS) that do provide facilities for viewing and navigating GIS images and there are also some systems that support enterprise information browsing. But there are a few systems for conveniently navigating and viewing both kinds of information. One of critical problems is how to develop effective interfaces for users to conveniently navigate and view both geographic images and enterprise information together. Also, to view GIS images/maps, current approach of using scrolling bars, windows (i.e. multi-views), *zoom-in* and *zoom-out* operations to show a huge GIS image is not convenient for the user's navigation, because it is difficult to obtain whole and detailed views of a huge image.

This paper presents some design issues for an enterprise geographic information system, focusing on effective interfaces that support GIS image navigation, detailed local image/map viewing, and enterprise information browsing.

Enterprise Information

The term *enterprise information* here refers to the information of those organizations that have geographically distributed branches, including each branch's resources (e.g. personnel) and product sales records. These information can be used for decision support on establishing (or closing) a branch or restructuring a branch (e.g. increasing / decreasing personnel) based on the population and the product sales record in a region.

Layout and Navigation

We use a graph for GIS image navigation. Each node in the graph represents a local area (e.g. a suburb, a town). Each edge between two nodes represents a relationship between two areas; it may be a neighborhood relation (e.g. two suburbs) or a hierarchic relation

(e.g. a town and a suburb). Each node is also related to the enterprise information in a region. The intention of using the graph for GIS image navigation is to give the user a global view of a huge GIS image and whole enterprise information.

We developed an enterprise GIS user interface with three underlying views: *a graph navigation view*, *an image layout view,* and *an enterprise information view.* The graph navigation view shows a sub-graph corresponding to the current user's focused image displayed in the image layout view. A node representing this image is in the central position of the sub-graph. The other nodes in the sub-graph represent this image's connection areas (e.g. neighbourhood) which may not be displayed currently in the image layout view. This sub-graph can give the user a good guidance for GIS image navigation. The navigation view displays GIS images/maps. The enterprise information view shows enterprise information for a specific application (e.g. a big organisation). The links among three kinds of views have been set up.

Mark Images

If we want to set up an object enterprise by selecting a place, we need to investigate similar object enterprises around the selected region to analyse whether it is necessary for establishing the object enterprise at that place. We use this place as a central point to find a sub-GIS map. The image mark for the object enterprise is used for searching all related mark images in a special region and the relations among these mark images are set up. That is we use the mark image as a pattern in searching similar object enterprises around selected region. To further explore similar object enterprises around this region, we can navigate this graph by selecting a node in this graph to change the focus point. After searching all surrounding areas by selecting the nodes in the graph, we can obtain the information of surrounding enterprises

A Systematic Approach to Plan Inspection Process

Ambrish K Srivastava, Gopesh Sharma
Tata Consultancy Services
Tulsi Ganga Complex
19-C, Vidhan Sabha Marg
Lucknow 226001
India
Ambrish_srivastava@lko.tcs.co.in,
gopesh_sharma@lko.tcs.co.in

Dr. Gargi Keeni
Tata Consultancy Services
SDF Building, 3rd floor
Salt Lake Electronics Complex
Block EP & GP, Sector V
Bidhan Nagar, Kolkata 700 091, India
gkeeni@mumbai.tcs.co.in

Inspections of software development work products have become an integral part of software quality improvement. However, experience tells, that inspection effectiveness (i.e. it's capability to find a defects) and its cost effectiveness vary significantly across organizations or even more striking, from one project to another in the same organization.

Some managers possess an intuitive latent understanding of the requirements for a better management of inspection, and make decisions using the skills and their experience. However, when wrong decisions are made and implemented into an inspection process, disastrous results may occur, increasing the cost of finding the defect at the later stages of testing or even delivering the product with lower quality level.

One of the most challenging and significant avenues of research in software engineering discipline is the investigation of how to ensure software quality, reduce the development cost and keep software projects within the schedule. Software inspection is practical approach to tackle all three issues. However, there still exist challenging questions, which software organization needs to be able to answer:

1. What is the most cost effective variation?
2. How to assign resources to an inspection in an optimal way?
3. How does the number and experience of inspectors influence software inspection?
4. The percent of error found during inspection?
5. How much to inspect?

In this paper an attempt is made to develop a ' Mathematical model for the software inspection process, which captures various factors, that have an impact on the effectiveness and cost of the inspection process, and that can explain most of their variation.

The approach is to develop the model for the software inspection process by using the discrete defect removal model and incorporating in it, the features of the system dynamics modeling. The concept of system dynamics modeling allows variation to the input parameter of the model with respect to the system dynamics variables i.e. the experience level of inspectors, complexity level of work products etc.

The feedback loop shown in figure shows, how the factor expressed in causal model of software inspection process behaves dynamically.

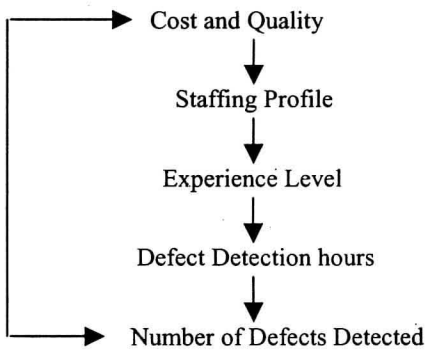


Figure 1: Feedback loop of Inspection Process

The feedback loop shown in figure takes into consideration

the amount of defect detection hours, which translate into number of defects detected, which in turn reflect the quality level achieved by inspection process. An inspector with less domain experience might spend more time on defect detection and will detects less defect as compared to inspector with high domain experience and hence more defect will escape to the next phase resulting in poor quality index and may result in at lower or higher cost value depending upon the cost of the inspector.

The model also accounts for the chances of defect detection of those defects, which escaped from previous phase, in to the next phase. Hence also considers the dependency of defect generated in the next phase with respect to inspection process effectiveness of the previous phase i.e. with the defect detected and the defect escaped in the previous phase. The model also account the chances of defect injection while closing the defects found at inspection, however in order to make the model simple it only been considered at the requirement phase of the software development process.

The model is basically based upon the inspection process followed in TCS, where the inspection process is divided into two broader categories:

1. Internal Quality Assurance
2. External Quality Assurance

At first level of inspection only the project team do the inspection among themselves with respect to the work product developed while at the second level of inspection quality assurance group assign reviewer from outside the project team to get a unbiased and second opinion on quality of developed work product.

The specification of the software inspection process model is the accumulation of knowledge gathered from the literature and the industry regarding the cause-effect relationships that exist in the model and impact on cost and quality level achieved.

An executable model requires that metrics be to be collected and entered into the model, in order to simulate a project. The metrics required by the model, define a metric set for collection by any software development organization, wishing to begin a process improvement program. Once collected, the metric values may be entered into the model and "played back" for analysis. The reward for collecting this set of metrics can result in more effective software inspection process.

The following parameters are required for calibration of the model:

- Inspection Effectiveness (of inspector at each experience level)
- Defect injection rate for various phases of software inspection process
- Average requirement, design defect amplification
- Chances of detection of defect escaped from previous phase.

The model assume uniform defect injection and detection rate. The above-discussed parameters generally will vary from organization to organization and also from project to project within an organization.

The model is general enough to incorporate any development process i.e. the model parameters are flexible enough to interface the model with any kind of development process model.

The model may be converted to a management simulator, providing researchers and software project managers with the ability to perform controlled experiments on their development environment

This model can be used to train the software project manager to develop skills for allocation of inspection effort, taking into consideration the various factors, which affect the quality and the cost of inspection. Using several projects of different complexity level, and iteratively experimenting with different inspection allocation, the trainee can learn the tactics and skills needed for software inspection effort allocations.

These contributions provide the foundation for further study of inspection process improvement and their impact on the cost and quality level achieved

A Software Complexity Measurement Based on C++ Source Code for Reverse Engineering

Jong-Wan Kim, Sung-Yul Rhew
Soongsil University
#1, Sangdo 5-dong, Dongjak-ku
Seoul, Korea
wany69@hitel.net, syrhew@comp.ssu.ac.kr

Sung-Eun Lee
Dong Seoul College
#423 Bokjung-dong, Sujung-ku, Sunggnam
Kyonggi, Korea
selee@haksan.dsc.ac.kr

ABSTRACT

For the last a decade, many methods for Software Complexity Measurement have been and published various methods for OO Software Complexity Measurement, recently. These methods have been used but most of reverse engineering is based on program analysis methods such as parsing code and data flows among functions or objects. If all source code is adjusted the same analysis method such as code parsing, we can't find major part in the system and same weight value will be put to subsystem. In this paper, class complexity was measured by new weighted method that check number of parameters, a return value and its' data type, to avoid the old methods' simplicity of checking only function's number. Moreover, to make class complexity measurement guideline for C++ source codes in reverse engineering steps we suggest measurement technique. That is, By weighting on class' interface, this paper shows practical result and adjusts new method for Complexity Measurement in C++ source codes constructed with object-oriented technology which called ECC(Enhanced Class Complexity)[1] Measurement Method.

INTRODUCTION

Measurement is fundamental to any engineering discipline, and software engineering is no exception. In practice, however, metrics are only used in hardware engineering; seldom do we see quantitative measurements in software. These are essential to management in measuring productivity and software quality.

A C++ implementation of a Server Secure System Project is tested and compared with CK and ECC (Enhanced Class Complexity).

PROPOSED I - TEST AREA AND CODE MAPPING

In an object-oriented system, minor test areas are implemented by a set of C++ functions F and a set of class C. Let MAP3 be a relation over F x NR. Table 1 shows an example of mapping from functions to minor test areas.

F x NR	*nr1*	*nr2*	*nr3*	*nr4*
f1	T		T	
f2		T	T	T
f3		T		T
f4	T		T	T

Table 1. Mapping from functions to minor test areas

Let MAP4 be a relation over C x NR, too. Table 2 shows an example of mapping from class to minor test areas.

C x NR	*nr1*	*nr2*	*nr3*	*nr4*
c1	T	T		
c2	T		T	T
c3		T		T
c4	T	T	T	T

Table 2. Mapping from class to minor test areas

Therefore, in this relation, minor test area *nr1* is set by {(f1, f4), (c1, c2, c4)}, *nr2*• •{(f2, f3), (c1, c3, c4)}, *nr3*

[1] Digital Sense Ltd. supported this research, and ECC method that is explained in this paper was adjusted in the Reverse Engineering project.

{(f1, f2, f4), (c2, c4)}, *nr4* {(f2, f3, f4), (c2, c3, c4)}. c4 is common between *nr1*, *nr2*, *nr3* and *nr4*, but there isn't enough space to put c4, we separate it in each part.

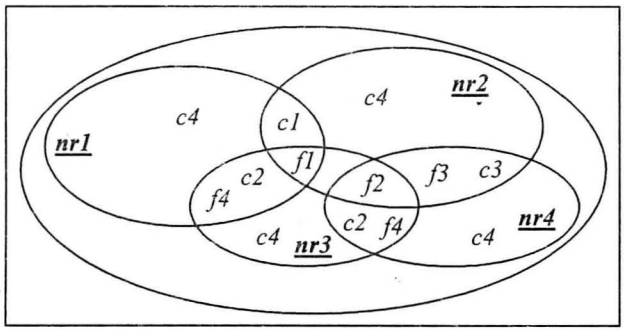


Figure 1. The relationship between minor test areas, functions and classes

PROPOSED II - ENHANCED CLASS COMPLIXITY

ECC_1(Enhanced Class Complexity) is not full ECC in this step, after; it will be changed to ECC. WMC, defined by Chidamber & Kemerer, is comprised the sum of weighted method complexity in a class without measure internal complexity of a function. That is, WMC has a simple summation of functions in a class. And we suggest another factor *cv* for ECC. *cv* is comprised the number of class variables.

$$ECC_1 = WMC + cv$$

Class complexity according to ECC_1 is the sum of the number of instance variables in a class and the sum of the static complexity of all local methods (functions).

The number of function's parameters and data types measure function complexity. Return value and data type are the base line, too. We consider the call by value and call by reference when the function is called. The weight value is put '1' in primitive data type, although the value is primitive, if it is pointer variable, we assign value 3. See the Table 3. We want to put the LOC, but it is not effective.

	Primitive Var	Pointer Var
Parameter	1	3
Return value	1	3

Table 3. The weighting values for primitive variables and pointer variables

Thus, ECC_1 is comprised such as next expression. *fv* is the sum of function parameters and return value.

$$ECC_1 = WMC + cv + fv$$

According to the table 1 and table 2 , let's assume that class c1 is mapping into function f1, class c2 is mapping into function f2. In the next expression, *cf* is the number of union set between class' *nr* and function's *nr*.

f1 = {nr1, nr3}, f2 = {nr2, nr3, nr4}, c1 = {nr1, nr2}
c2 = {nr1, nr3, nr4}, that is, f1 $\in$ c1
cf = n(f1 $\bigcup$ c1) = 3
ECC = WMC + cv + fv + cf

The number of classes and functions that belong to minor part is expressed frequency of use in a subsystem. Thus, analyzer can select and first analyze, preliminarily, the minor part for reverse engineering.

CASE STUDY IN PROJECT

See the full text.

CONCLUSIONS

In this paper, we tested the source of our project with improved measurement method ECC than C & K's method. Also, the complexity measurement was used as a method to verify a complexity and a system dependency of analysis target source in reverse engineering and can select a more important functions or classes using ECC.

The future study, which must be researched more, is that calculates comparative complexity between each classes and functions. If do this, we can get more detailed analysis and reliability of the system.

An Automatically Generated Runtime Environment for Testing the Functionality of EJB Components

Jihyun Lee and Gyu-Sang Shin

S/W Contents Technology Department, ETRI Computer & Software Technology Lab.
161 Kajong-Dong, Yusong-Gu, Daejon, 305-350, Republic of Korea
{jihyun, gsshin}@etri.re.kr

Abstract

In this presentation, I will address an Automatically Generated Environment for Verifying the functionality of EJB components (AGEVE). The reliability of a newly generated component or an assembled composite component is acquired from functionality verification. The interface of an EJB component is used as a communication entity to invoke functions and receive execution results. Our system automatically generates verification clients and provides the clients to component developers for testing functions of interfaces by themselves.
Our client is deployable software that is automatically generated with analyzed EJB information. A set of decomposable functions is expressed in the client and the functions are executed with parameter instances that are created at runtime without manual coding or compilation. The verification environment is designed to execute functions with user requirements to be known at runtime. Our approach relieves developers of the need to test by automatically providing a self-testing environment.

1. Introduction

Because of the fast changes in the software market, the need for reusable and maintainable software has been increased. A component-based development method has been introduced as a solution for that need. This method produces the following representative models: Enterprise JavaBeans (EJB) by Sun, CORBA Component Model (CCM) by OMG, and COM and DCOM by Microsoft. The EJB model provides an architecture technology that builds reusable and maintainable enterprise applications. The EJB model is used in our system.

The EJB component is deployable unit software that is composed of classes and interfaces. The implementation of business functionality is represented in classes and the functionality of the components is specified in interfaces. Because the interfaces of the components are used for communicating with other components, it is necessary for all component users to ensure that functions of the components are correctly executed. In general, component developers make client programs to execute functions and check results, and they frequently change the programs to verify other functions or functions with other parameters.

In the AGEVE, the clients are automatically generated with analyzed EJB information. The clients take runtime requests, execute user-selecting functions, and create parameter instances at runtime. The AGEVE enables component developers to easily verify component functions when a new component is generated or a new component-assembled application is being built.

2. Component Analysis to Automatically Generate Verification Clients

An EJB component is composed of interfaces, beans, and a deployment descriptor. The interface is classified into a home interface and a remote interface. The home interface defines methods that create, find, and remove an enterprise bean. The remote interface defines methods that a client may call. The bean can be implemented in business methods and is compiled into a bean class file. One bean has particular business functions that are specified in home interfaces and remote interfaces. The deployment descriptor defines structural and assembly information that is not included in the EJB codes and is expressed in XML. The EJB component is packaged into a JavaARchive (JAR) file for deployment. In our system, we use the deployment descriptor of the EJB as a measure to find the component functionality.

We get reflected method signatures for beans after analyzing package names of the deployment descriptor. A Method signature is defined with a method name, parameter types, and a return type. The set of method signatures from each bean becomes a test set in our EJB verifying environment. The method signatures of a bean can be decomposed into a unit method.

3. Functionality Verification by Executing Functions with Runtime Requests

Our verification steps are composed of analyzing an EJB component and showing the analyzed functionality information, selecting a bean with special functionality, generating verification clients, invoking a method, and validation execution results.

Our EJB verifying client supports the component developers in creating parameter instances that are set with frequently changed values at runtime. The EJB verifying client reports execution outputs and internal errors after developers run one method by themselves. Even complex or user-defined data type parameters can be created with the values inputted by developers. The parameter instances can be stored in serialized files and our verifying client supports the reuse of the files in executing the methods with the same parameter type.

The EJB client is a java application to call functions of a bean. Our verification clients support dynamic execution and are generated using skeleton templates. The skeleton templates are written in JavaServer Pages (JSP) containing Hyper-Text Markup Language (HTML) and Java logics. The skeleton templates are changed with analyzed component information and runtime user requests. The newly changed templates are automatically provided to component developers and are used for testing functionality of an EJB component.

4. Conclusions

Our automatically generated environment for verifying the functionality of EJB components generates verification clients and allows component developers to execute methods by themselves. Modifications of EJB clients happen frequently to component developers. Hand writing mistakes often occur when client codes are changed. Our automatically generated clients allow developers to avoid hand writing errors and extra compilation processes. In addition to that, the developers can control the scope of functionality testing in one bean or in one method without designing testing programs in advance.

With the AGEVE, a newly generated component and a newly assembled composite component can be verified in view of functionality. The AGEVE is executable through self-deploying if an EJB component is provided. Therefore, the AGEVE can be reused for verifying interfaces and functions of components. The complex and the user-defined data type instances are created and are managed for reuse.

In our verification environment, the generated clients are minimum and our clients can create instances of complex or user-defined data type parameters and set their values at runtime. Our verification environment is tested with J2EE, Weblogic, WebSphere, and JEUS application servers. Our verification environment shows better execution support and better client generation than that of an existing commercial product. The AGEVE was integrated in our COmponent-BAsed deveLopment Tool (COBALT) that was developed for component construction and assembly.

The automatically generated verification environment lessens the burden on component developers and the reliability of each component is acquired more easily.

Object Oriented Programming in Spreadsheet

Jong-Myung Choi
Soongsil Univ.
Seoul, Korea
jmchoi@comp.ssu.ac.kr

Spreadsheet Programming

- Features
  - Very popular
  - Used by end-users and experts
- Merits
  - Easy to learn
  - Easy to use
- Drawbacks
  - High error rates
  - Low abstraction
  - Low reuseability
  - Difficulties in maintenance

2

Object-oriented Programming

- Features
  - Very popular
  - Used by experts
- Merits
  - High reusablility
  - High abstraction
  - Easy maintenance
- Drawbacks
  - Hard to learn

3

Motivation

- Motivation
  - Combine the spreadsheet programming and object-oriented programming
- Requirements
  - Should be easy to learn.
  - Should be easy to use.
  - Should support object-oriented programming.
  - Should support spreadsheet programming.

4

Ideas

- How to define classes
  - Adopt XML to struture data
    - Easy to learn
    - Easy to use
- How to define operations
  - Adopt formula in text
  - Adopt flowchart in graphics
- How to use classes and objects
  - Provide spreadsheet GUI
  - MVC architecture

5

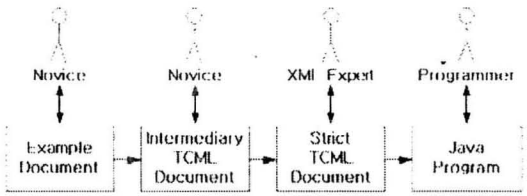
Object-oriented Spreadsheet

- Object-oriented Spreadsheet
  - Class Designer
    - Define classes
      - XML - member fields, methods
      - Flowchart - methods
  - Object Spreadsheet
    - Uses classes
    - Creates objects
    - Calls methods
    - Interacts with users

6

Data Modeling

- Interaction between users and the class designer



7

TCML

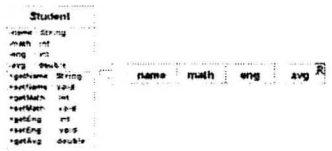
- TCML DTD

```
<!ELEMENT class (field*, method*)>
<!ATTLIST class
        abstract    NMTOKEN #IMPLIED
        name        NMTOKEN #REQUIRED
        kindof      NMTOKEN #IMPLIED>
<!ELEMENT field (#PCDATA)>
<!ATTLIST field
        name        NMTOKEN #REQUIRED
        type        CDATA   #IMPLIED
        number      NMTOKEN #IMPLIED
        readonly    NMTOKEN #IMPLIED>
<!ELEMENT method (#PCDATA)>
<!ATTLIST method
        name        CDATA   #REQUIRED
        type        CDATA   #REQUIRED
        args        CDATA   #IMPLIED>
```

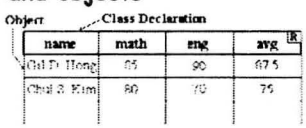
8

Object Spreadsheet

- Class Declaration



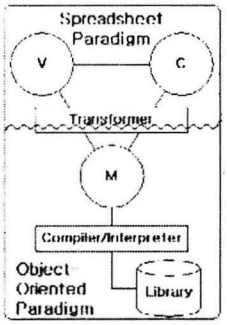
- Class and objects



9

System Architecture

- Architecture



10

Object-Oriented Spreadsheet

- Class Designer
- Spreadsheet

11

Conclusions

- Object-oriented Spreadsheet
  - Allows object-oriented analysis, design, and programming
  - Allows distributed computing
  - Reduce errors
  - Provides high-level abstraction and modularity
  - Easy to maintain
  - Easy to learn
  - Easy to use

12

GIGA: Graphical Definition of Production Rules in a Spatial Parser Generator

Hiroaki Kameyama, Kazuhisa Iizuka, Buntarou Shizuki, and Jiro tanaka
Institute of Information Science and Electronics
University of Tsukuba
1-1-1 Tennoudai, Tsukuba, Ibaraki, 305-8573, Japan
kame@iplab.is.tsukuba.ac.jp

INTRODUCTION

Visual language means a language that uses figures in addition to text. Visual languages are used in various fields, such as ER diagrams and OMT object diagrams. Spatial parser generators automatically generate parser of visual languages by providing their grammars. For visual systems, analyzing visual languages is not enough. Actual visual systems must execute statements according to the result of the analysis and redraw the visual sentence preserving the semantic relationships between figure elements.

We previously developed the spatial parser generator Eviss[1] [4][5], which automatically generates a spatial parser by defining the grammar of the visual language. In Eviss, we introduced "action" into Constraint Multiset Grammars (CMGs) [2][6] in order to express the behavior of visual programming system. By providing just the grammar and the action of a visual programming system, we can easily describe a visual system. The defined grammar is analyzed by Eviss, which automatically generates the parser.

However, in Eviss, the input of the grammar is performed using text. If all composition elements and action performed are given in textual form, it is difficult to understand what kinds of figures compose the grammar and what happens when the grammar is analyzed. If we can use a figure for inputting the grammar and edit it directly, the grammar will become easier to understand.

SPATIAL PARSER GENERATOR

Extended Constraint Multiset Grammars

We use Extended CMGs [1] in order to define the grammars of visual systems. We have extended the original CMGs to include action A defined as "script program executed when the production rule is applied." In the Extended CMGs, we can specify arbitrary actions, such as computing values and rewriting figures.

The production rule of Extended CMGs is defined as follows:

$$
\begin{aligned}
T(\vec{x}) \quad ::= \quad & T_1(\vec{x}_1), ..., T_n(\vec{x}_n) \quad where \\
& exists \quad T'_1(\vec{x'}_1), ..., T'_m(\vec{x'}_m) \\
& not\_exists \quad T''_1(\vec{x''}_1), ..., T''_l(\vec{x''}_l) \\
& where \quad C \quad and \quad \vec{x} = F.
\end{aligned}
$$

In this paper, we describe the computation tree (Figure 1), as an example.

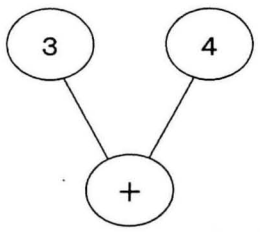


Figure 1: Computation tree.

The computation tree is defined recursively by the following two production rules.

1. A non-terminal symbol "Node" consists of a circle and a text in the center of it.
2. A non-terminal symbol "Node" consists of a circle, a text string, two nodes and two lines. The two nodes are connected to the circle by the lines.

The production Rule 1 defines a node that represents a number; production Rule 2 defines a node that represents an operator.

GRAPHICAL DEFINITION OF RULES

We have newly developed a successor to Eviss called GIGA system. GIGA's graphical interface for the CMG Input Window has two screens in addition to the CMG input Window of Eviss (as shown in Figure 2). i.e., a pre-action screen and a post-action screen. The user inputs the figures into the Pre-action screen before the action is performed. The user edits the figures in the Post-action screen after the action is performed. Operations in each screen can be performed similarly to a general drawing editor.

The user defines the grammar using the graphical interface for the CMG input window as follows.

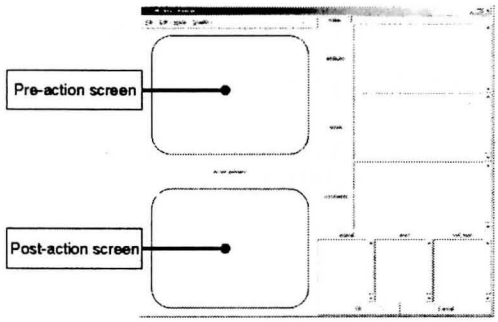


Figure 2: Graphical interface for CMG input window.

1. The user inputs the figures to be analyzed on the Pre-action screen.
2. GIGA duplicates the figure input to the pre-action screen to the post-action screen.
3. On the post-action screen, the user modifies the duplicated figure to show the result after the action is performed.
4. GIGA infers the production rule from two screens and outputs it to the respective definition part in textual form.
5. If required, the textual form can be edited.

Computation tree example

Rule 2 of the computation example can be defined as follows. We draw a circle, a text, two lines and two nodes in the pre-action screen. GIGA then infers the components and outputs them to the Component definition part (shown in Figure 3). We next delete two nodes and two lines in the post-action screen to define the action of the production rule. GIGA infers the action from the difference of the pre-action screen and the post-action screen, and then outputs it to the Action definition part.

CONCLUSION AND FUTURE WORKS

In this paper, we described the GIGA system in which a user can define the grammar using figures. GIGA has two screens for defining the grammar for inferring the production rule from the difference of the two screens. GIGA outputs the resulting production rule by comparing the two screens. With GIGA, the user can define the grammar easily and can understand the grammar visually.

REFERENCES

1. Akihiro Baba, Jiro Tanaka. "Eviss: A Visual System Having a Spatial Parser Generator," In proceedings of 1998 Asia Pacific Computer Human Interaction, IEEE Computer Society Press, pp. 158-164, Jul 1998.
2. Sitt Sen Chok, Kim Marriott, "Automatic Construction of User Interface from Constraint Multiset Grammars," In Proceedings of 1995 IEEE Symposium on Visual Languages,IEEE Computer Society Press, pp. 242-249, Sep 1995
3. Allen Cypher, David Canfield Smith, "KidSim: End User Programming of Simulations," In Conference proceedings on Human factors in computing systems, Denver, Colorado, United States, ACM Press/Addison-Wesley Publishing Co, pp. 27-34, May 1995.
4. Kenichiro Fujiyama, Kazuhisa Iizuka and Jiro Tanaka, "VIC:CMG Input System Using Example Figures," In Proceedings of the international Symposium on Future soft-ware Technology, Najing, China, pp. 67-72, Oct 1999.
5. Kazuhisa Iizuka, Jiro Tanaka, and Buntarou Shizuki, "Describing a Drawing Editor by Using Constraint Multiset Grammars," In proceedings of the international Symposium on Future soft-ware Technology, Zhen Zhou China, pp. 119-124, Nov 2001.
6. Kim Marriot, "Constraint Multset Grammars," In Proceedings of the 1994 Symposium on Visual Languages, IEEE Computer Society Press, pp. 118-125, Oct 1994.
7. Kakuya Yamamoto, "3D-visulan: A 3D Programming Language for 3D Applications," In Proceedings of Pacific Workshop on Distributed Multimedia Systems, the Hong Kong University of Science and Technology, pp. 199-206, Jun 1996.
8. Kakuya Yamamoto, "Visulan: A Visual Programming Language for Self-Changing Bitmap," In Proceedings of International Conference on Visual Information Systems, Victoria University of Technology (in cooperation with IEEE), pp. 88-96, Feb 1996.

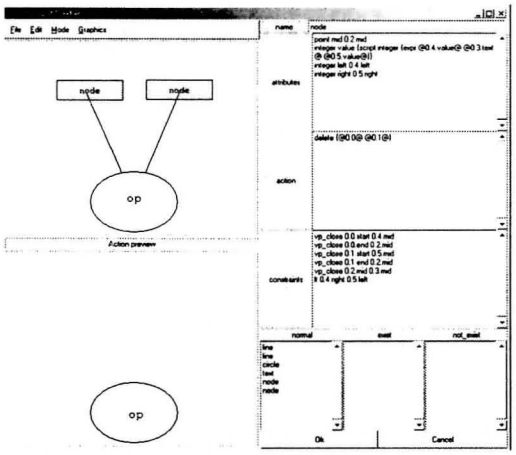


Figure 3: Definition of production rule 2.

E-commerce and ontology

Jianming Yong[1 2] Jun Yan[2]
[1]Department of information Systems
Faculty of Business
University of Southern Queensland
QLD4350, Australia
Email: yongj@usq.edu.au

[2]CICEC - Centre for Internet Computing
and E-Commerce
School of information technology
Swinburne University of Technology,
PO Box 218, Hawthorn, Melbourne, Australia 3122

ABSTRACT

Internet has brought e-commerce to a brand-new stage. Internet-based e-commerce is attracting more and more researchers' interests. How to effectively develop and manage these Internet-based e-commerce systems has become a critical issue for the future of e-commerce as a whole. This paper discusses some important technologies related to Internet-based e-commerce, XML and ontology. Through XML and ontology, we will demonstrate how to build and organize the ontological repositories over the Internet.

KEYWORDS

XML, e-commerce, architecture of e-commerce, ontology

INTRODUCTION

OVERVIEW OF XML-BASED STANDARDS FOR E-COMMERCE

ebXML

OAG

BizTalk

cXML

Open Verticals

XBRL

XEDI and XML/EDI

FinXML

Bolero$_{XML}$

ONTOLOGY OF INTERNET-BASED E-COMMERCE

E-commerce applications from ontology

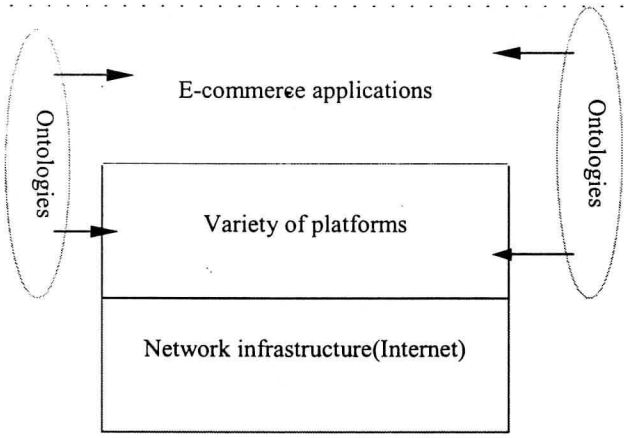


Figure 1 ontologies and e-commerce

Topology of ontology

ARCHITECTURE OF INTERNET-BASDED E-COMMERCE

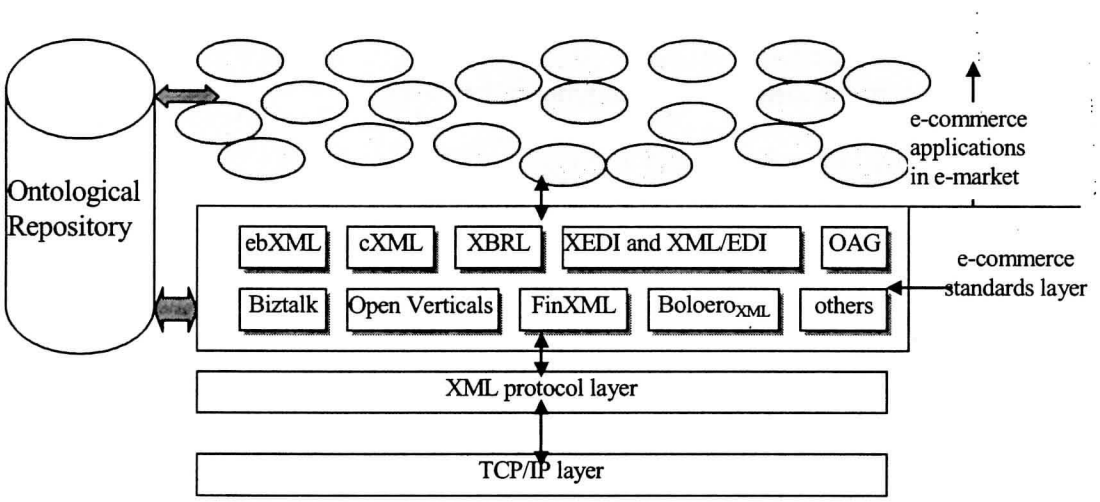


Figure Architecture of Internet-based e-commerce

CONCLUSIONS AND FUTURE WORK

ACKNOWLEDGEMENT

REFERENCES

This page is intentionally made blank

Brief reports from Session Chairs

Session 1A

SooYong Park chaired the session 1A. S1A-1 was not presented, other two papers were presented by the first author according to SooYong Park after the session.

-shioya

Session 1B

Kazunori SHIOYA was chaired the session 1A. The presented paper is S1B-1 only. Presenters of S1B-2 and S1B-3 were no show therefore not presented.

-shioya

Session 2A and 5B

I was the session chair of session 2A and 5B. All speakers gave their presentation at that time.

In the session 2A, Prof. Tanaka from Tsukuba Univ. was a char for my presentation 2A-1 instead of me.

--

Session #, Name, Affiliational country

2A-1 Kei Kurakawa, Japan

2A-2 Shu Fengdi, China

2A-3 Prof. Park, Korea

--

5B-1 Chen Guoning, China

5B-2 Hao Zhong, China

5B-3 Hye-Jue Eun, Korea

- Kei Kurakawa

Session 2B

no report from the session chair, however, I was attended 2B session. S2B-1, S2B-3 and S2B-4 papers were presented by the first author and co-author. S2B-2 paper author no showed was not presented by anybody.

-shioya

Session 3A

In the session 3A(?), Chaired by me, all 3 papers were presented.

Dehua Ju, ASTI Shanghai

Session 3B

All three papers of Session 3B were presented in English.

By the way, since my paper erroneously appeared in two sessions in the program, I went to another session for quite a while. When I found I should chair and go to Session 3B, the session had begun. I missed part of the presentation of 3B-2.

Y. S. Kuo

Session 4A

Thank you for your all the efforts for the symposium. I think it was really successful.

All the expected presenters presented their papers in the session.

Shusaku Iida

Session 4B

All three papers were presented for session 4B.

For 2 of the 3 papers it was the second author who presented the paper.

The authors were all on time before the session started, and all the session (presentations, questions, etc.) went well. The presentations did not last the whole 30 minutes (they were rather short, about 20 mins each) and the whole session (4B) was covered in about 60 minutes.

The presenters were quite professional and did a good job. I really enjoyed chairing session 4B.

Miguel A. Serrano

Session 5A

It was nice to see you in Xi'an and Wuhan and I really enjoyed the conference.

I chaired the session 5A: Formal Methods, System Evolution and Maintenance (2)

All the three authors:

Mr. Ozaki, Mr. Noh, and Mr. Iida

presented their papers in English.

Jaejoon Lee

Session 5B

-> See the report of Session 2B

Session 6A

It is my pleasure to congratulate you on successful completion of ISFST2002. It was nicely managed. Also I would like to say a lot of thanks for your cooperation throughout the process from paper submission to end of ISFST2002 program.

Regarding your mail, I was the chair of session 6A. Here is the report of this session.

Paper Presented: 6A-1

Paper 6A-2 Absent

Paper 6A-3 Absent

Total three paper were there in session 6A. Only one was presented as reported above and the other two presenters were absent.

Nazir A. Zafar

Session 6B

I am Yoshitaka MATSUMURA, and was the session chair of 6B.

Fortunately all the papers of 6B, 6B-1, 6B-2 and 6B-3 were presented.

Yoshitaka Matsumura

Session 6C

The first paper was NOT presented. The second and the third paper were presented.

S6C-1 Research of XML Wrapper System based on XML/CORBA

Xiu-fen Fu - Not presented

S6C-2 Huilin - Presented

S6C-3 Eun Sook - Presented

Jiro Tanaka

---

*My presented session number was **3C-5. (Poster session)***

I do not know my paper 1011C have twice listed in ISFST2002 program booklet till receive your E-mail. So I do not presented twice.

Fu Xiufen

Session 7A

In the session 7A,

first two papers(7A-1, 7A-2) were presented, but the last paper(7A-3) was NOT presented.

Doo-Kwon BAK.

Session 7B

I was chairing the session 7B and the following three papers were presented:

7B-1, 7B-2, 7B-3

Sebastian Ng

Session 7C

I was chair of the 7C session.

Fortunately we had three presenters in this session.

Cheers!

--- bear

Summary from a Program Co-Chair

The PC meeting at Lushan in August accepted 49 papers, there were 68 (46 submitted to Japan, 18 submitted to China, and 4 late papers) papers were rated based on referee reports and judged. 47 final papers were submitted by the final due (2 dropped). At the symposium in Wuhan, presented technical papers were 41 finally. Locally corrected 79 abstract papers were not assigned to the referees because of too short to be judged, however, Wuhan local strongly requested to invite them to the symposium, so we decided to recommend these authors to join the poster sessions.

The listed posters in the locally made "program booklet" were 107 and assigned into two parallel poster sessions. There were much duplication of papers between technical and posters. Also, most of the poster presenters were no show, as audiences may notice. This made some confusion among the presenters and local staffs. For instance, session 2S which I had attended shortly was a perfect Chinese session presented and communicated in only Chinese, only 2 paper out of 8 were presented and some joined from other session, even from next session 3D, This is chaotic situations. The poster program attached in this after proceedings is a "planed as is" version that means nobody checked which presentations were presented or not.

Since there are no clear agreement or discussion for the posters at the Lushan PC meeting, because of "time out" strongly requested from Wuhan local, it looked that nobody checked and managed these poster sessions officially. So, actual number of presented papers were not counted therefore unknown. To avoid this in next symposium, the poster session should also be programmed and managed for the next symposium. ISFST staff listing should be included in the "program booklet" as in the ISFST-2002 official Web and it should be up to date.

The session chair assignment of this year was made by local staff and was "random" assign regardless of their entry. It should be assigned with acknowledgement (OK) from chair candidate with explanation of session management rule at least prior to start and the chairs should report after their session to the PC chairs. We should discuss these issues at least by or/and at the PC meeting. This is the program committee's duty.

Unfortunately, we failed this year both for advertisement and management. Absences of the Russian speakers are very sorry, however, we have attendees from 7 regions, there are main land China, Taiwan, Macau, Korea, Japan, Australia, USA and new from Mexico, attendees are expanding. There are many attractive places for study, visiting and sight seeing in China. We could expect more attendees might come next symposium, if we propose more attractive program and provide handy information of the venue area. So the planning and managing properly is the key to success the symposium. Let us make next ISFST-2003 in Xi'an to be more successful!

Kazunori SHIOYA

A Program Co-Chair of ISFST-2002

Summary from a General Co-Chair

First of all I deeply appreciate all of you who participated ISFST2002 and cooperated with ISFST2002. We have done it successfully though we have met some issues to be solved every day. But it's over and it became to be our good memory.

Here I'd like to write summary of this symposium as one of general chairs. It would be useful for the next that will be held in Xi'an in 2003.

At first I consider about the framework of ISFST2002. There are mainly three parts, the first one is Tutorial in Xi'an, the second one is Symposium and the third one is optional tours. It's very nice to provide good chances for ISFST2002 participants to join different activities. Especially it's great to have over one hundred participants at Tutorial in Xi'an who came from outside from China and China. We could exchange and communicate many things through these three parts of ISFST2002 not only technical issue but also our culture and national backgrounds and economics. It's very useful to understand each other maturely.

Secondly I show the scale of this symposium. The participant of the Tutorial was about 100. The total number of ISFST2002 symposium participants is 104. There were 50 participants from outside China such as Sweden, Mexico, USA, Australia, India, Pakistan, Korea, Macao, Taiwan, Japan and 54 participants from China. It's a good size to discuss deeply. It's interesting to show that 90% of participants are from academic and the only 10% of participants are from industries. Especially in China over 95% of participants are from Universities. The people from industries are mainly from India and Japan.

Thirdly I describe the brief history of ISFST. We started this symposium to exchange software technologies between China and Japan in Shanghai in 1987. The hosts of symposia were SEA Japan and some affiliations in China for example Shanghai Software Center and Beijing University. We have held symposia every year at Chinese cities in principle. We changed this symposium name as "**** International CASE Symposium (xICS)" and called for participants from all over the world especially focus on Asia countries from 1991. For 'CASE' technology was very popular in the Software Engineering area at that time. We change the name of symposia to ISFST in 1997. For 'CASE' technology has already downed and we have to extend the scope of symposia wider and wider to cover many new activities and technologies.

You may understand the outline of ISFST now. Finally I describe the meaning of ISFST from my experiences and perspective. All information technologies have been owing to USA from the beginning of the computer invention including hardware and software. We Asian people have been always merely users of the information technologies. But time is changing. The information technologies began to cover not only civilization but also our culture such as our daily life. The civilization means social institutions like automatic control of mass production, control systems, commercial systems, banking systems, communications including inter-net and etc. The culture

means our spiritual activities like learning, thinking, many kinds of art such as music and movies, literatures, inter-communication among countries and etc.

There need strong connection between the information technologies and culture when we cover our culture with software and computers. There are many cultures in Asia, because there are many races in Asia. They have their languages and cultures. We have to invent our information technologies to understand our Asian cultures each other. We can't import it from outside of Asia. So we have to understand each other among Asian people focus on the future software technologies. ISFST has been providing a good chance to exchange many things. I call it as 'DIGITAL SILK ROAD for Asian people'. Heavier the traffic is, better 'DIGITAL SILK ROAD is. It was proved as 'SILK ROAD' in Tang Dynasty about 1500 years ago. I am convinced that new things will be created in the place where the traffic be beyond heavy. Let's make such situations.

We expect that new ideas, new technologies, new young researchers, new relationships among countries, new business chances will be brought by ISFST in the future. It's my hope and dream.

Thanks again for all participants and all staffs of ISFST2002.

I hope to see you again next year in Xi'an.

Cheers.

Akira Bear Kumagai

A General Co-Chair of ISFST-2002.

The Proceedings (summaries) of ISFST-2002

ISBN 4-916227-14-X (CDROM)

Published in December 2002 by

Software Engineers Association

3-12 Yotsuya, Shinjuku-ku
Tokyo 160-0004, Japan
Tel: **+81-3-3356-1077**
Fax: **+81-3-3356-1072**
sea@sea.or.jp
http://www.sea.jp/

ソフトウェア技術者協会

Software Engineers Asociation

ソフトウェア技術者協会 (SEA) は，ソフトウェアハウス，コンピュータメーカ，計算センタ，エンドユーザ，大学，研究所など，それぞれ異なった環境に置かれているソフトウェア技術者または研究者が，そうした社会組織の壁を越えて，各自の経験や技術を自由に交流しあうための「場」として，1985年 12月に設立されました．

その主な活動は，機関誌 SEAMAIL の発行，支部および研究分科会の運営，セミナー／ワークショップ／シンポジウムなどのイベントの開催，および内外の関係諸団体との交流です．発足当初約 200人にすぎなかった会員数もその後増加し，現在，北は北海道から南は沖縄まで，500 余名を越えるメンバーを擁するにいたりました．法人賛助会員も 24社を数えます．支部は，東京以外に，関西，横浜，名古屋，九州，広島，東北の各地区で設立されており，その他の地域でも設立準備をしています．分科会は，東京，関西，名古屋で，それぞれいくつかが活動しており，その他の支部でも，月例会やフォーラムが定期的に開催されています．

「現在のソフトウェア界における最大の課題は，技術移転の促進である」といわれています．これまでわが国には，そのための適切な社会的メカニズムが欠けていたように思われます．SEA は，そうした欠落を補うべく，これからますます活発な活動を展開して行きたいと考えています．いままで日本にはなかったこの新しいプロフェッショナル・ソサイエティの発展のために，ぜひとも，あなたのお力を貸してください．

ＳＥＡＭＡＩＬ Ｖｏｌ．１３，Ｎｏ．９－１０ ２００３年1月１５日発行 編集人 岸田 孝一
発行人 ソフトウェア技術者協会（ＳＥＡ）
〒１６０-０００４ 東京都新宿区四谷３－１２ 丸正ビル５Ｆ
T: 03-3356-1077 F: 03-3356-1072 sea@sea.or.jp
印刷所 有限会社 錦正社 〒１３０-００１３ 東京都墨田区錦糸町４－３－１４
定価 １，０００円 （禁無断転載）

ソフトウェア技術者協会
〒160-0004　東京都 新宿区 四谷 3 - 12 丸正ビル 5F
Tel: 03 - 3356 - 1077　Fax: 03 - 3356 - 1072
E-mail: sea@sea.or.jp
URL: http://www.iijnet.or.jp/sea