Computer Science and Information Systems

Published by ComSIS Consortium

Volume 8, Number 3
June 2011

Computer Science and Information Systems

Vol 8, No 3, June 2011

ComSIS is an international journal published by the ComSIS Consortium

ComSIS Editorial office:
University of Novi Sad, Faculty of Sciences,
Department of Mathematics and Informatics
Trg Dositeja Obradovica 4, 21000 Novi Sad, Serbia
Phone: +381 21 458 888; **Fax:** +381 21 6350 458
www.comsis.org; Email: comsis@uns.ac.rs

Volume 8, Number 3, 2011
Novi Sad

Computer Science and Information Systems

ISSN: 1820-0214

ComSIS Journal is sponsored by:

Ministry of Education and Science of Republic of Serbia - http://www.mpn.gov.rs/

Computer Science and Information Systems

AIMS AND SCOPE

Computer Science and Information Systems (ComSIS) is an international refereed journal, published in Serbia. The objective of ComSIS is to communicate important research and development results in the areas of computer science, software engineering, and information systems.

We publish original papers of lasting value covering both theoretical foundations of computer science and commercial, industrial, or educational aspects that provide new insights into design and implementation of software and information systems. ComSIS also welcomes surveys papers that contribute to the understanding of emerging and important fields of computer science. Regular columns of the journal cover reviews of newly published books, presentations of selected PhD and master theses, as well as information on forthcoming professional meetings. In addition to wide-scope regular issues, ComSIS also includes special issues covering specific topics in all areas of computer science and information systems.

ComSIS publishes invited and regular papers in English. Papers that pass a strict reviewing procedure are accepted for publishing. ComSIS is published semiannually.

Indexing Information

ComSIS is covered or selected for coverage in the following:

- Science Citation Index (also known as SciSearch) and Journal Citation Reports / Science Edition by Thomson Reuters, with 2010 two-year impact factor 0.324,
- Computer Science Bibliography, University of Trier (DBLP),
- EMBASE (Elsevier),
- Scopus (Elsevier),
- Summon (Serials Solutions),
- EBSCO bibliographic databases,
- IET bibliographic database Inspec,
- FIZ Karlsruhe bibliographic database io-port,
- Index of Information Systems Journals (Deakin University, Australia),
- Directory of Open Access Journals (DOAJ),
- Google Scholar,
- Center for Evaluation in Education and Science and Ministry of Science of Republic of Serbia (CEON) in cooperation with the National Library of Serbia,
- Serbian Citation Index (SCIndeks),
- doiSerbia.

Information for Contributors

The Editors will be pleased to receive contributions from all parts of the world. An electronic version (MS Word or LaTeX), or three hard-copies of the manuscript written in English, intended for publication and prepared as described in "Manuscript Requirements" (which may be downloaded from http://www.comsis.org), along with a cover letter containing the corresponding author's details should be sent to one of the Editors.

Criteria for Acceptance

Criteria for acceptance will be appropriateness to the field of Journal, as described in the Aims and Scope, taking into account the merit of the content and presentation. The number of pages of submitted articles is limited to 30 (using the appropriate Word or LaTeX template).

Manuscripts will be refereed in the manner customary with scientific journals before being accepted for publication.

Copyright and Use Agreement

Computer Science and Information Systems

Volume 8, Number 3, June 2011

CONTENTS

EDITORIAL

It is our great pleasure to announce that Computer Science and Information Systems has been included in the 2010 release of Journal Citation Reports by Thomson Reuters, and received the two-year impact factor of 0.324. We would like to use this opportunity to gratefully acknowledge all former Editors-in-Chief of ComSIS: the late founder of ComSIS Prof. Branislav Lazarević (University of Belgrade, Serbia), Prof. Vladan Devedžić (University of Belgrade, Serbia), and Prof. Ivan Luković (University of Novi Sad, Serbia), as well as the current and former members of the Editorial Board, the Editorial Council, and the Managing Board, and finally all reviewers, authors, and everyone who contributed to achieving this significant milestone for our journal.

This second regular issue of ComSIS in 2011, and third issue overall, consists of five regular articles and 14 papers selected from ISCSCT 2010, 3rd International Symposium on Computer Science and Computational Technology, held during September 14–15, 2010, in Jiaozuo, China, and ISECS 2011, 4th International Symposium on Electronic Commerce and Security, held during June 24–26, 2011 in Shanghai, China. We thank the organizing committees of both events, and especially the guest editor for this issue, Fei Yu (Peoples' Friendship University of Russia), for the contribution of quality articles to this issue of our journal.

The first regular article "Building Computers in Serbia: The First Half of the Digital Century," by Jelica Protić and Dejan Ristanović, reflects on the history of Serbian digital computer development by describing the early work on CER-10, the first digital computer built in Serbia, honoring its inventors, surveying the CER family of computers which stemmed from CER-10, and analyzing the effect it had on the personal computer revolution in Serbia and the evolution of the information society that can be seen today.

The next regular paper, entitled "Workflow Specification for Interaction Management Between Experts in a Cooperative Remote Diagnosis Process," by M. A. Laredj and K. Bouamrane, proposes a workflow model to assist cooperative maintenance, with the goal of coordinating the interactions between various intervening actors in the process of diagnosis and maintenance of detected breakdowns.

In "Access Control Framework for XML Document Collections," Goran Sladić, Branko Milosavljević, Zora Konjović and Milan Vidaković present the eXtensible Role-Based XML Access Control Framework (XXACF), a framework for controlling access to XML documents in different environments

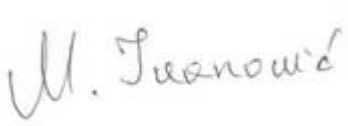
which enables the definition of context-sensitive access control policies on different priority and granularity levels, the enforcement of access control for different operations on XML documents, and different ways of access-control enforcement for the same operation.

The article “Feature Diagram Formalization Based on Directed Hypergraphs” by Miguel A. Laguna, José M. Marqués, and Guillermo Rodríguez-Cano, proposes the use of hypergraphs to integrate the structure and additional constraints of existing approaches to specifying feature diagrams into a unique characterization, in order to accurately represent the variability and commonality of software product lines.

Finally, “Integrating Instance-level and Attribute-level Knowledge into Document Clustering,” by Jinlong Wang, Shunyao Wu, Gang Li, and Zhe Wei presents a document clustering framework which incorporates instance-level knowledge in the form of pairwise constraints and attribute-level knowledge in the form of keyphrases. The framework incorporates metric learning with pairwise constraints and learning of two kinds of knowledge by combining distance-based and constraint-based approaches.

On behalf of the Editorial Board and the ComSIS Consortium, we would like to acknowledge the efforts of the authors in preparing and submitting their high-quality contributions, and also the diligence and hard work of the reviewers which went into the preparation of this issue of Computer Science and Information Systems.

Editor-in-Chief
Mirjana Ivanović



Managing Editor
Miloš Radovanović

GUEST EDITOR'S MESSAGE

This issue of Computer Science and Information Systems contains revised and expanded versions of selected quality papers presented either at the Third International Symposium on Computer Science and Computational Technology (ISCSCT 2010) or at the Fourth International Symposium on Electronic Commerce and Security (ISECS 2011).

The first conference, ISCSCT 2010, took place on September 14–15, 2010, in Jiaozuo, China, and was co-sponsored by Henan Polytechnic University, China; Peoples' Friendship University of Russia, Russia; Feng Chia University, Taiwan; Zhengzhou University, China; Fudan University, China; South China University of Technology, China; Nanchang Hang Kong University, China; Jiaxing University, China; and Academy Publisher of Finland.

ISECS 2011 took place on June 24–26, 2011 in Shanghai, China, and was co-sponsored by Fudan University, China; Peoples' Friendship University of Russia, Russia; South China University of Technology, China; Feng Chia University, Taiwan; Henan Polytechnic University, China; Nanchang Hang Kong University, China; Jiangxi University of Science and Technology, China; Qingdao University of Science & Technology; and Academy Publisher, Finland.

This issue of ComSIS includes a collection of 14 papers selected from the 527 submissions to ISCSCT 2010 and ISECS 2011. All the papers included in this issue have gone through a rigorous peer-review and revision process for their originality and quality.

The first paper has the title "Code Cache Management based on Working Set in Dynamic binary translator" and the authors are Ruhui Ma, Zhichen Ni, Erzhou Zhu, Kai Chen, and Haibing Guan. The paper presents two novel code cache management schemes, SCC (static code cache) and DCC (dynamic code cache) based on working sets, which adapt to software code cache, compared to conventional replacement policy, such as Flush and LRU, etc.

The paper "Data Extraction and Annotation based on Domain-specific Ontology Evolution for Deep Web" by Chen Kerui, Zuo Wanli, He Fengling, Chen Yongheng, and Wang Ying presents a robust, highly efficient data annotation method based on ontology evolution. A simpler ontology which can significantly improve annotating efficiency is defined. The dynamic ontology adopted in this paper can evolve concurrently with the process of data

annotation. In this way, the new ontology will work more accurately and sufficiently for new web annotation. Experiments indicate that this method can improve the accuracy and efficiency of data extraction and annotation.

The paper “Extracting Minimal Unsatisfiable Subformulas in Satisfiability Modulo Theories” by Jianmin Zhang, Shengyu Shen, Jun Zhang, and Sikun Li presents a depth-first-search algorithm and a breath-first-search algorithm to compute minimal unsatisfiable cores in SMT, adopting different searching strategy. They also report and analyze experimental results obtained from a very extensive test on SMT-LIB benchmarks.

The paper “Indexing Temporal Information for Web Pages” by Peiquan Jin, Hong Chen, Xujian Zhao, Xiaowen Li, and Lihua Yue presents several alternatives to integrate the inverted index and MAP21-tree together to answer keyword queries with temporal constraints. Experimental results about five alternatives were analyzed and presented. The topic discussed in the paper is a very interesting issue.

The paper “Parallel processing on Block-based Gauss-Jordan Algorithm for Desktop Grid” by Yizi Shang, Zhizhong Liu, and Ling Shang analyzes all the possibilities for parallelism in the block-based Gauss Jordan algorithm, while former research works just focused on intra-step based parallelism. Then a formal description on data dependence between operations is made and some regulations of data dependence on different operations are summarized. According to those regulations and features of desktop grid environments, this paper proposes a new parallel programming paradigm for desktop grid systems. Experiments testify that the algorithm proposed in this paper can better adapt to volatile environments.

The paper “Problem Solving by Soaking the Concept Network” by Xixu Fu and Hui Wei presents a solution for reasoning with dense knowledge. By soaking the knowledge network, the proposed process is discovered and represented as a reasoning graph.

The paper “Research on Discovering Deep Web Entries” by Ying Wang, Wanli Zuo, and Yulan Qi presents a new framework WFF for efficiently locating domain-specific deep web databases based on focused crawling and ontologies, by constructing a Web page classifier (WPC), form structure classifier (FSC), and form content classifier (FCC) in a hierarchical fashion.

The paper “Study of Privacy-Preserving Framework for Cloud Storage” by Ruwei Huang and Xiaolin Gui presents a privacy-preserving cloud storage framework to resolve the following problems: the data organization structure, the generation and management of keys, data retrieval, the treatment of change of users’ access rights and dynamic operations of data, and the interactions among participants.

The paper “SVM Based Forest Fire Detection Using Static and Dynamic Features” by Jianhui Zhao, Zhong Zhang, Shizhong Han, Chengzhang Qu, Zhiyong Yuan, and Dengyi Zhang describes a method to detect forest fires from video images. In the light of rapidly changing weather patterns, the proposed method has great utility. The proposed method is technically sound and has great promise.

The paper “The Trustworthiness Analyzing of Interacting Business Process Based on the Induction Information” by Xianwen Fang, Changjun Jiang, and Zhixiang Yin presents a behavior trustworthiness analysis method for business process based on induction information. The contributions of the authors have two aspects. One is that the behavior relativity analyzing method is proposed based on Petri nets at the static interacting stage. The other is that the behavior conformance is studied based on K-bounded process-mining methods at the dynamic running stage.

The paper “Voice Activity Detection Method Based on Multi-valued Coarse-graining Lempel-Ziv Complexity” by Huan Zhao, Gangjin Wang, Cheng Xu, and Fei Yu presents a novel robust method for voice activity detection (VAD) that is based on multi-valued coarse-graining Lempel-Ziv Complexity (MLZC), which is an improved algorithm of the binary coarse-graining Lempel-Ziv Complexity (BLZC). In addition, authors use the fuzzy c-Means clustering algorithm and the Bayesian information criterion algorithm to estimate the thresholds of the MLZC characteristic, and adopt the dual-thresholds method for VAD.

The paper “Worst Case Performance Bounds for Multimedia Flows in QoS-enhanced TNPOSS Network” by Ke Xiong, Shenghui Wang, Yu Zhang, Zhifei Zhang, and Zhengding Qiu presents a lower E2E delay bound for QTNPOSS networks by using network calculus theory, where the inherent properties (e.g., packet length and peak rate) of the flow are taken into account.

The paper “A Two-Tiered Reliable Application Layer Multicast” by Xinchang Zhang, Meihong Yang, Guanggang Geng, Wanming Luo, and Xingfeng Li presents a two-tiered reliable application layer multicast protocol called HRALM. The protocol uses a domain-based clustering solution to build the ALM tree, and the paper presents a feasible retransmission approach.

The paper “Multi-Scale Image Semantic Recognition with Hierarchical Visual Vocabulary” by Xinghao Jiang, Tanfeng Sun, and GuangLei Fu presents a hierarchical semantic model which organizes multi-scale semantics from the semantic space. The hierarchical semantic model is used to organize the multi-scale semantics into a level-by-level structure and to define the relationship between the semantics.

I would like to give our great thanks to the reviewers for their helpful comments and all of the authors for their contributions, efforts and

enthusiasm. Thanks are also due to the ComSIS Consortium, and especially to the Editor-in-Chief of ComSIS, Mirjana Ivanović and other staff in the Editorial Office for their advice and help in making the publication of these selected papers possible. Finally, my gratitude goes out to Paul Werbos (National Science Foundation, USA), Gary G. Yen (Oklahoma State University, USA), Derong Liu (Chinese Academy of Sciences, China), and Jun Wang (Chinese University of Hong Kong, Hong Kong), for their vital roles in the organization of ISCSCT 2010 and ISECS 2011 events.

Guest Editor
Fei Yu
Peoples' Friendship University of Russia

DOI: 10.2298/CSIS110505021P

Building Computers in Serbia: The First Half of the Digital Century

Jelica Protić[1] and Dejan Ristanović[2]

[1] University of Belgrade, School of Electrical Engineering,
Bulevar kralja Aleksandra 73
11000 Belgrade, Serbia
jeca@etf.rs
[2] PC Press,
Osmana Đikića 4
11000 Belgrade, Serbia
dejanr@pcpress.rs

Abstract. In this paper, we describe the early development of CER-10, the first digital computer built in Serbia, honor its inventors, and follow the professional path of its chief designers, prof. Rajko Tomovic and prof. Tihomir Aleksic, who became the first university professor of computer engineering in Serbia. We also give a short overview of CER family of computers that were developed after the CER-10 till mid-seventies. In the early eighties, computer revolution continued with personal computers, so we show the early attempts to produce this kind of computers in Serbia, from "build it yourself" campaigns to the industrial production, and we analyze implications of these attempts to the development of user community and evolution to the information society.

Keywords: history of computing, CER family, personal computers, computer industry.

1. Introduction

During the Second World War, the interest to speed up computation was driven by the need to decipher codes and to run ballistics calculations. A group of women called "human computers" were employed to use differential analyzer, a mechanical analogue computer designed to solve differential equations at The Moore School of the University of Pennsylvania in Philadelphia. But on the backstage, John W. Mauchly and J. Presper Eckert were developing ENIAC, the first computer that could perform ballistics calculations faster by a factor of 1,440. ENIAC was composed of 18,000 vacuum tubes, 70,000 resistors and 10,000 capacitors. It had 40 panels, weighed 30 tons, and it was powered by 150 kilowatts. The first operators of the machine were six women [1].

In 1996, ENIAC turned 50, as well as the leading organization of computer professionals and scientists ACM, which organized a year-long celebration, with multiple conferences and activities which honored the first 50 years of technological accomplishments that lead to today's information society [2]. The celebration took place in the USA 14 years before the anniversary in honor of the first half-century of digital computers in Serbia. For many IT professionals half-younger then CER-10, the first computer developed and built in Serbia, it was a perfect occasion to find out about computing pioneers, the first electrical computing machinery, the first programmers and professors who set the foundations of computer science and engineering in Serbia.

The central event of the celebration organized by Informatics Society of Serbia, Institute "Mihajlo Pupin", University of Belgrade-School of Electrical Engineering, Ministry of Telecommunications and Information Society, and Chamber of Commerce of Serbia, was the exhibition set in the hall of National bank of Serbia [3]. About 70 exhibits, such as old computers, peripherals, photos, panels and publications from early sixties to the present time, helped the older visitors to remember, and the young students to research the roots of our profession and its impact on the technological development and industrial productivity of Serbia. It also demonstrated the enormous changes in our everyday life in which not only the vision of a computer on every desk and in every home, but also a smart device in every pocket, transformed the way how we live, teach and learn.

In the following chapters we will examine the roots of computing in Serbia, present CER-10 computer, honor its inventors, and provide the overview of the whole CER family. The second part of the paper is devoted to the personal computers of the 1980s that were produced in Serbia: Galaksija (Galaxy), Lola 8, TIM computers, and Pecom 32/64. Finally, we will elaborate on the process of choosing school computer, and the transition to the production of PC compatible computers in 1990s.

2. Before the Beginning: The Roots of Computing between Electrical Engineering and Mathematics

The transition from mechanical computing devices to electrical computers required the knowledge and experience in electrical engineering and mathematics. The first university level lecture in the area of electrical engineering in Serbia was held in 1894 by Professor Stevan Marković, the first lecturer and founder of Electrical Engineering Chair at the Great school in Belgrade, which was the third of that kind in Europe [4]. Telegrams about this event were sent to already famous electrical engineers living abroad: Nikola Tesla and Mihajlo Pupin in New York and Vojislav Marinković in Paris. Only four years later, Professor Marković also founded electrical engineering laboratory at the basement of Kapetan Miša's Palace (now the building of the University of Belgrade). Since then, electrical engineering has been studied

in the Great School, and later at the University of Belgrade which developed from it. The first diplomas in electrical engineering were issued in 1922.

One of the best friends of Professor Marković and his son's best man was the famous mathematician, Mihajlo Petrović Alas. The beginning of the story about "Computing in Serbia" and the exhibit number one (Figure 1) at the exhibition [3] is Petrović's hydro integrator that he constructed, and presented at the Exposition Universelle in Paris in 1900, and received the Gold Medal for it. The hydro integrator was based on the principle of hydraulic analogy, and it could solve two classes of differential equations. In addition, it was a kind of predecessor of the plotters, because calculation data was written automatically with a pen on the paper rolled around the cylinder [5].

Fig. 1. Mihajlo Petrović's hydro integrator

3. Beginning of the Digital Century: CER-10

Until the early sixties, only six countries in Europe succeeded in developing their own electronic computers: Great Britain, France, Germany, Poland, Russia and Yugoslavia. The Yugoslav product CER-10 (in Serbian: Cifarski Elektronski Računar – Digital Electronic Computer) was a vacuum tube, transistor and relays based computer [6][9]. From 1956 till 1960, a team of 70 engineers, programmers, technicians and specialists were involved in its construction at the Institute Boris Kidrič - Vinča in Belgrade. The project was initiated by academician Dušan Mitrović. The machine was designed by professor dr Tihomir Aleksić, academician Rajko Tomović and their associates: Ahmed Mandžić, Petar Vrbavac, Vukašin Masnikosa, Dušan Hristović, and Milojko Marić [37].

After initial prototype's testing in Vinča, the construction continued in the Institute “Mihailo Pupin” in Belgrade, where the machine was extended with the so-called Statistical Unit, so by the end of 1962 the project was finished and CER-10 could be moved to the SKNE-DSUP (The Federal Commission for Nuclear Energy-State Secretariat for Internal Affairs) building, which later belonged to TANJUG (Telegraph Agency of the new Yugoslavia). CER-10 was used for statistical cryptologic processing of information for the Yugoslav Federal Government, including SSUP (Federal Secretariat for Internal Affairs) and TANJUG, as well as for mathematical problems solving, related to the scientific projects in SKNE (The Federal Commission for Nuclear Energy) in Vinča.

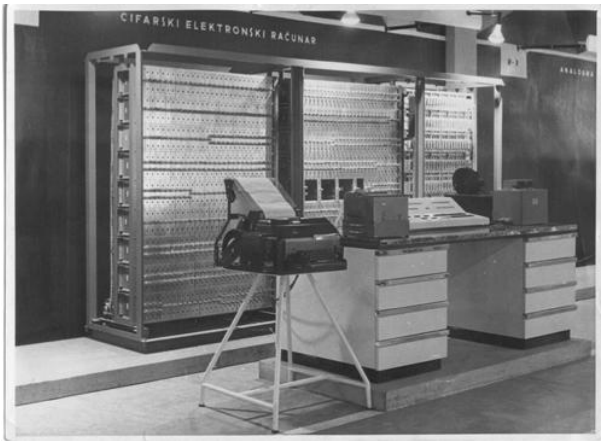


Fig. 2. CER-10 at the Belgrade Technical Fair, August 21-31 1960, Belgrade

CER-10 [9] was the universal one-address machine with the average processing speed of about 50,000 simple operations per second and about 1,600 additions per second. Data in CER-10 was represented in multiples of 5-bit binary groups, so the operations could be performed on 5-bit, 15-bit or 30-bit binary words. Instruction was composed of 25 binary digits. A 5-bit opcode, which could potentially code 32 instructions, was used for only 17 codes of regular instructions (Table 1) and additional 7 instructions which were later introduced by building a statistical unit. Index field was used to determine if the operation should be performed on 5-bit, 15-bit or 30-bit binary words. 15-bit address field determined the address of the lowest 5-bit group of the operand.

The main memory was organized in two subsystems of matrix ferrite memory cores Philips 6D3 pfi 2mm and switching cores 6E2. The memory word was 30+1 bits, and it could store up to 6 characters: numbers, letters or symbols coded with 5 bits each. Access time for the ferrite memory was 10 microseconds.

The control unit of the CER-10, and its arithmetic unit were composed using the standard logic circuitry modules made of electronic vacuum tubes,

transistors, Ge-diodes and R, L, C discrete components. As an input unit, the machine used Photoelectric Reader of the punched paper tapes Ferranti type TR 2B (the speed was 300 char/sec). For output, it used the Paper tape Puncher Creed type 25 (max. 100 char/sec). The Siemens teletype T-100 with printing speed of 8-10 char/sec was also used.

The Power Supply system was an independent motor generator with the nominal power 20/15,5 KVA, produced by the Rade Končar company, Zagreb. The rectifiers with three phase circuitry had Si-diodes, types 14R2 and 10R2 (Th. Houston). There was the automatic regulation, the relay protection and the signalizations for all power units. The size of computer room in the TANJUG building was 80 m^2, with double flooring and air conditioning. The metal rack dimensions were 2 * 2 * 0,70 meters for each of the seven rack units.

In building the CER-10, the following components were used: Philips electronic tubes of types ECC 81, EL 83 etc. (approx. 1.750 pieces); Transistors 2N396, OC 76, OC 44 (1.500 pieces); Ge-diodes OA 85 Philips, for logic circuitry, (approx. 14.000 pieces); Electronic relays type Schrack (approx. 650 pieces); Pulse transformer core D25 (approx. 1.700 pieces); Delay pulse Lines (approx. 850 pieces), etc.

Table 1. Instruction set of CER-10 [9]. The first letter defines the statement, **n** represents the memory address, **Z** represents the index (Z, X or Y) which determines the size of operand, A stands for accumulator, B and C are special registers, and **n** should be interpreted together with the index. s/X/ represents the value stored in X.

Statement	Functionality	Comment
AnZ	s/A/→n without delete of s/A/	Store A to n
RnZ	s/A/→n with delete of s/A/	Store A, Delete A
CnZ	logical and of s/A and s/n→A	A and n→A
JnZ	s/B/→n without delete of s/B/	Store B to n
K	end of program execution	Stop
Nn	if s/A/>0 go to n	Conditional jump
On	if s/A/<0 go to n	Conditional jump
Pn	if s/A/≥0 go to n	Conditional jump
RnZ	s/A/- s/n/→A	A-n→A
SnZ	s/A/+s/n/→A	A+n→A
TnZ	s/n/→C	Load C from n
DnZ	s/A/ x 2^{-a}; a= s/n/	Shift right A,n
LnZ	s/A/ x 2^{a}; a= s/n/	Shift left A,n
MnZ	s/C/ x s/n/→A and B	Mul AB,C,n,
QnZ	s/C/ / s/n/→ B	Div B,C,n
I	Output the lowest 5 bits of s/A	Out A
U	One character from the tape+s/A→A	Add A, Input

In the interview [12], Vukašin Masnikosa, the member of the team that developed CER-10, explained that the programming was done in machine

language and that the CER-10 proved its usability by decoding an important radio-message sent to Egypt from Israel, so President Josip Broz Tito sent the messenger who traveled by airplane to deliver the message personally to President Nasser.

In March 2006 the Institute Mihailo Pupin donated the case and parts of the CER-10 to the Museum of Science and Technology in Belgrade, where it will be displayed as a valuable exhibit.

4. Designers of CER-10 and their Heritage in Serbian Computing Education

The CER-10 implementation was based on the design of Professor Tihomir Aleksić and academician Rajko Tomović, who worked with a group of 70 associates, programmers and technicians. The two of them later became professors at the Faculty of Electrical Engineering in Belgrade, and set the foundations of computer engineering studies in Serbia. Professor Rajko Tomović (1919-2001) greatly contributed to the design of the first analog and hybrid computers built in the Vinča Institute from 1950 till 1960. After that, he joined the Institute Mihajlo Pupin, where he started the Laboratory for Robotics. In 1962 he became associate professor, and in 1964 full professor of automatic control at the Faculty of Electrical Engineering. He was the member of Serbian Academy of Sciences and Arts (SANU), and the designer of multifunctional hand prosthesis, so-called Belgrade hand, exhibited in the Museum of Robotics in Boston [7][8].

Professor Tihomir Aleksić (1922-2004) was the first elected university professor of computer engineering in Serbia. He got his Ph.D. in the field of switching theory in 1958, and started his research at the Mathematical Laboratory of the Vinča Institute, where he worked till 1960, when he transferred to the newly founded Institute for Automation and Telecommunications, Belgrade, the predecessor of the Mihajlo Pupin Institute. Professor Aleksić designed the laboratory prototype of a bookkeeping computer, a prototype of electronic printer, the electronic teleprinter, the telephone switching system, and other devices that have announced the digital age in this region [7].

His academic career started by the promotion to invited full professor at the Technical Faculty of Niš in 1967. After the establishment of the Faculty of Electronics in Niš, he became its first Dean. During the same year, the Faculty of Electrical Engineering of the University of Belgrade started the major changes in its study programs, by introducing integrated 5-year engineering studies. The curriculum contained a new module called Data Processing, and professor Aleksić's course Systems for Data Processing was one of the major subjects. In 1971, he became a full professor at the Faculty of Electrical Engineering in Belgrade, and the founder of the Department of Data Processing, which was the nucleus that started the development of the present Department of Computer Engineering and Informatics, today's

teaching bases of two major curricula in computing: Computer Engineering and Informatics as well as the Software Engineering.

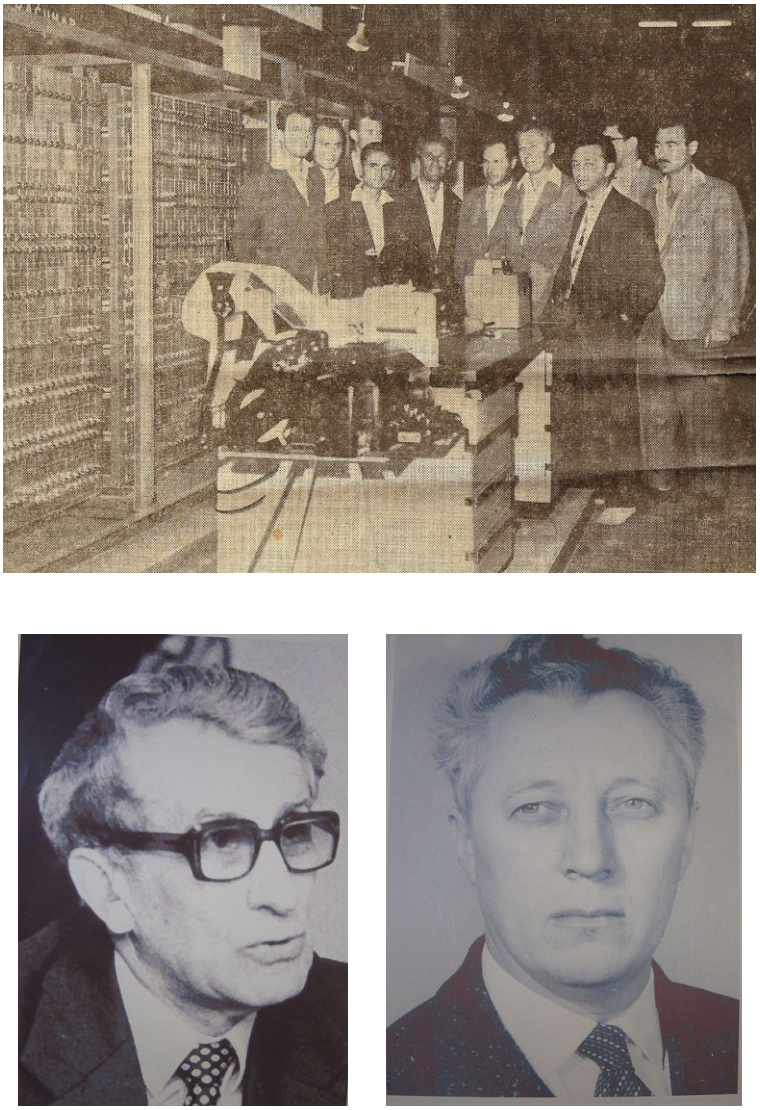


Fig. 3. The CER-10 Team, in "Borba" newspaper from September 6, 1960, and photos of professors Rajko Tomović and Tihomir Aleksić

5. CER and HRS families: the Successors of CER-10

After the construction of CER-10, Institute Mihajlo Pupin developed and produced the whole family of CER computers (CER-11, CER-22, CER-200 CER-12), which have been applied for data processing in multiple enterprises, banks and federal agencies [10][11]. They also developed specialized CER computers for the Yugoslav Army from 1965 to 1989. The last CER computer remained functional in the Vojvođanska Bank-Zrenjanin until the late 1980s, when it was replaced by the VAX minicomputer. An overview of CER family is summarized in Table 2.

While CER-12, CER-22 and CER-200 were mainly used for bookkeeping and banking applications, CER-30, described in [29] was something completely different. It was a predecessor of programmable electronic calculators, designed by professor Nedeljko Parezanović, who made an original multiplication and division algorithms, dr Milojko Marić, and dr Bogdan Janković. Due to the lack of components, the building of this machine was delayed, but the problem was solved when Professor Tihomir Aleksić provided germanium transistors produced in Ei-Niš. Instead of a display, CER-30 used 15 light indicators, and the keyboard was replaced by doorbell switches. However, the machine was used for a long time for financial computations at the institute.

The period from late 1960 until 1970 was marked by the design, development and production of hybrid computer systems HRS-100, intended for scientific and technical research, modeling of complex dynamic systems in real and accelerated time scale, and to effectively solve a wide range of scientific tasks in the following fields: power engineering, space industry, medicine, chemical technology, electronics and process automation.

The HRS-100 was ordered by and made in cooperation with the Institute for problems management of the USSR Academy of Sciences. Bringing together the world's achievements in the field of analog and digital computer technology with a range of original solutions, this system achieved the capabilities and performance of the world's top-class computers of the time. The HRS-100 has been implemented in 1971 in the VLSI technology and represented the modern system of the third generation with many original solutions (the electronic control of coefficients for hybrid computing, the system of interrupts, a hybrid loop for graphics, the dynamic order of priorities, the link device with the high precision AD and DA convertors, etc.).

The main program orientation continued from 1974 until 1990, which brought an expansion of the development program to the special purpose computer systems, as well as the design of real-time control applications. It was the time of intensive development and implementation of hybrid computer systems. Three HRS-100 systems were delivered to the institutes in USSR in Moscow and Novosibirsk. Another six hybrid mini-computers were also produced for faculty laboratories in Sarajevo, Skopje, Belgrade, Zagreb, Split and Subotica.

Table 2. An overview of CER family [11]

Model	Designed (End of exploitation)	User companies	Remarks
CER-10	1956/1960 (1966)	SSUP, TANJUG, ETŠ	the first electronic computer constructed in Serbia
CER-11	1965/1966 (1988)	JNA-DSNO	the mobile military computer
CER-20 CER-30	1963/1965 (prototypes)	IMP, Ei, RIZ(TRS)	CER-30 was the programmable electronic calculator, using original algorithms for multiplication and division designed by Nedeljko Parezanović
CER-22	1966/1968 (1975)	Beobanka, Jugopetrol, Belgrade Watersupply and Wastewater	production run: 3 printed circuits with transistors
CER-200/ 202/ 203	1966/1971 (1985)	Planika-Kranj, PIK-Tamiš, GIK-Banat, Kanal DTD-Novi Sad, Mehano-grafija, Jugopetrol, VMA, EI Poljobanka, IMP	production run: 15 printed circuits with transistors
CER-12	1969/1971 (1997)	Cred. Bank Zrenjanin, ERC-IMP	production run: 2 third generation 57 machine instructions, CDC disks, AMPEX magnetic tape unit, VLSI circuits, wire-wrapped Augat boards
CER-101-Kosmos	1972/1973 (1988)	JNA-VTI	the mobile military computer, third generation VLSI circuits, wire-wraped Augat boards, magnetic drum VERMONT 1004
CER-111	1974/1975 (1988)	JNA-DSNO	the mobile military computer third generation 57 machine instructions VLSI circuits, wire-wraped Augat boards

6. Personal Computers of the 80s

The CER project, however fascinating it was, could not retain the momentum necessary to keep up with the developments of the global computer industry. During the sixties and the seventies, Serbian companies bought and utilized computer systems produced by IBM, Honeywell, and DEC. In the early eighties, another computer revolution was approaching - home computers were ready for their big time [38]. So the creativity of Serbian engineers aroused once again.

"We can achieve the Computing Revolution only if we have a domestic computer" [15]. This sentence from an editorial of the first publication about personal computers released in the former Yugoslavia "Computers in your home," sounded logical at the time, at least as much as today it sounds funny. But in the early 80s it was important to produce a domestic TV, domestic appliances, a domestic airplane and, why not, a domestic home computer.

Fortunately, in the early eighties the domestic computer design was not a mission impossible. The computer hardware and software were simple enough that they could be designed, produced and tested by small teams, without expensive resources, on a moderate budget. In global terms, the first home computers were created in garages and constructed by the young people, who did not always have formal qualifications.

There were many garage built computers, but only some of them, due to the inventive design, quality software, and a bit of good fortune, could eventually reach the market. The situation on the global market at the time was strange: a dozen of mutually incompatible computers competed for customers, ready to invest endless time, enthusiasm, and a lot of money into the machines which could not do a really useful job.

At the time, Yugoslav economic system was inert, new products development was slow and often dependent on political decisions. The private companies were not only discouraged, but prohibited by the law. Production of computers required imported chips, and the possibility to import them was limited, as a person entering the country could only import small items worth about 50 Deutschmarks. Companies had to prove that their imports were covered by their exports, and to buy foreign currency at a rate significantly higher than the official.

But every cloud has its good: the unreasonable policies, along with many other factors, resulted in an incredible event - a "build it yourself" computer was assembled by approximately ten thousand people. This, of course, was not exactly the first home computer, as a few iterations were necessary to reach the winning concept.

As in many other areas, it is not easy to tell who came first - the construction of a small computer was attempted by many enthusiasts, but no projects were brought to an end. Basically, most of the constructors tried to clone some of the foreign computers, with the possible simplification of hardware, and the use of original system software, namely the BASIC interpreter.

The first home computer that was manufactured in the form of a working prototype, presented at several tradeshows and in the media, was EL-82 [16]. It was designed by Voja Antonić and produced by "Elektronika inženjering" ("Electronics Engineering") in the city of Zemun, at the beginning of 1982. The EL-82 used the Zilog Z80A microprocessor, with user memory of 16 kilobytes (expandable to 64 KB) and ROM of 16 KB. The EL-82 was functionally similar to the TRS-80 Model 1, but it's hardware was completely different, designed with fewer integrated circuits, that made it more suitable for production, but not entirely compatible with the TRS-80. Microsoft Level II BASIC was stored in the ROM, featuring the possibility of operation with numbers in double precision (15 significant digits), alphanumerics, sound generation and low-resolution graphics.

It is interesting to note that the character generator also supported the Cyrillic alphabet (though the keyboard was Latin), so it was possible to enter the Cyrillic text. The ROM also contained monitor program that provided direct access to the computer's memory debugging and working with programs written in the machine language.

The EL-82 was announced as the computer with the ability to expand: the Centronics port was provided for the printer connection, and the possibility of later upgrade with the disc drive (instead of the classic tape recorder used for recording programs) was also mentioned. The other plans included the development of new programming languages for specialized purposes, connecting with industry controllers, etc. The computer was presented to the public during the year 1982 and in the first half of the 1983, after which it quietly disappeared. The company "Electronics Engineering" concluded that the computer production was too expensive - estimated cost of some 250,000 dinars, which was about 6,000 Deutschmarks at the time, was too much for potential customers. But fortunately, there was a better idea...

6.1. Galaxy (Galaksija)

Galaksija (Galaxy) computer was constructed in 1983 by Voja Antonić, who resigned from the "Elektronika Inženjering" and started his own business. He began to work on the project in July 1983, and the computer was ready for its first presentation in August. Galaksija was based on the Zilog Z80A microprocessor, the top of the 8-bit technology of that time. In the beginning it had 4 KB of RAM (up to the final launch, the memory was extended to 6 KB [18]), 4 KB ROM (with the possibility of extension with additional 4 KB ROM II), and used the monitor or black-and-white TV set that could display 32 * 16 letters and 64 * 48 pixels. The features of Galaksija were comparable to the Sinclair ZX-81 - the price was similar, but the Galaksija had an advantage of the professional keyboard, while the ZX-81 had more peripherals available and a lot of commercial and free software.

Fig. 4. The Galaksija Computer

Galaksija brought two original solutions. One of them was the software support for video, which simplified the hardware, and made the computer cheaper, which was regarded as particularly important. It also allowed for the single-layer printed circuit board, suitable for the "build it yourself" construction. The second characteristic was a BASIC interpreter that Voja Antonić developed by himself, taking some segments (e.g., the floating point arithmetic) of the TRS 80 Level 1 BASIC. The development and optimization of the interpreter took a lot of work, so the 4 KB ROM also contained the software support for video, the line editor, elementary manipulations with strings, a clock, and links for further expansion of the programming language. BASIC commands had shortcuts [19], for example, P. instead of PRINT, for easy typing and memory savings. The intention to unambiguously reduce all the commands to a single letter requested certain changes in the BASIC syntax, so some of the commands were renamed - for example, TAKE instead of READ or BYTE instead of POKE.

The Galaksija computer was offered in the "build it yourself" campaign announced in a special edition "Computers in your home" [20] (December 1983), the first publication on home computers issued in the former Yugoslavia. It happened eight years after the subscribers to Popular Electronics in the USA received the new issue in the mail, with a prototype of the Altair minicomputer on the cover [13]. Altair was named after the star mentioned in the TV show Star Trek [14], and its minicomputer kit was the first one that anybody could obtain for less than $400. Just like in the case of Altair, the magazine also played an important role in introducing Galaksija in Serbia (Figure 5).

It turned out that components for such a computer were affordable for an enthusiastic reader: a microprocessor, memory and other integrated circuits

for Galaksija could legally be imported from abroad by post, and the production of the printed circuit boards, the keyboards, and the case has been organized in Yugoslavia.

Fig. 5. Magazines that promoted "build it yourself" campaigns

In order to succeed in this campaign, it was necessary to spend several months in preparations and publications of articles in the magazine for popularization of science, "Galaksija", so in October 1983 the preliminary order forms appeared in the magazine [21]. The response has surpassed all the expectations: over a thousand readers of "Galaksija" expressed the desire to build their first computer. It motivated the editorial board to further efforts: Voja Antonić spent a lot of time improving computer's hardware and software, and the journal "Galaksija" was looking for the most appropriate way to organize the purchase of components, relying on (as then said) "small businesses". Companies "Mipro" and "Electronics" from Buje (now in Croatia), in collaboration with the Institute of Electronics and Vacuum Technique (Slovenia), supplied printed circuit boards, keyboards and masks, while "Mikrotehnika" from Gratz supplied the chips by mail orders. Editorial board of "Galaksija" collected orders and organized EPROMs burning.

Meanwhile, the commercial version of the Galaksija computer was offered, in the joint effort of "Electronics Engineering" and "Institute for learning and teaching resources", that decided to fund (already almost completed) development of Galaksija, in order to put it on the market and offer it to the schools in Yugoslavia. The price of 30,000 dinars was equivalent to about 600 Deutschmarks. After the release of the publication "Computers in your home" in December 1983, over 8,000 readers from all over former Yugoslavia ordered the Galaksija kit; in subsequent months, orders continued to arrive, deliveries were delayed, but finally all eager builders got their packages.

Due to the simple construction of Galaksija, the most of devices that readers built got to work without problems, and for those less fortunate, the service was organized. The commercial model arrived later than promised, but the computer gradually found its way to schools. Perhaps the most important thing was that the media enthusiastically supported the campaign: it was announced and talked about on the radio, television and in magazines, so it can be considered as a milestone that marked the beginning of computer revolution in former Yugoslavia.

In the publication "Computers in your home 2" a scheme for self-building of the memory expansion board, developed by Jovan Regasek, was presented [22], as well as three-channel tone generator and high-resolution graphics, designed by Nenad Dunjić and Milan Tadić [23]. A lot of software was developed for Galaksija, some of which was distributed in an unusual way - via radio waves, in cooperation with Zoran Modli and Radio "Belgrade 202" [24]. "The Institute for learning and teaching resources" distributed programs commercially, so Galaksija even got a chess-playing program - Ivan Gerenčir and Milan Pavićević first adapted a famous Sargon to the memory and graphics of Galaksija. ROM 2 [25] was developed by Voja Antonić - with this 4 KB extension, Galaksija got an assembler program, machine-language monitor program, several new BASIC commands (including floating-point functions), and the printer support.

The subsequent development of the Galaksija computer continued in the magazine "Svet kompjutera" ("Computer World"), where Nenad Balint, Vojislav Mihailović, Bojan Stanojević, and other associates managed to attract the attention of the users. "Computer World" realized the vision of founders of the entire operation: the gathering of the readers who turned into the programmers. They predominantly published games: Diamond Mine, Squash, Light Cycle Race, Blade Alley, Inspector Spiridon, etc. These programs were written in machine language, which used the potentials of Galaksija to its maximum. The crown of the whole project was the construction of Galaksija Plus (Galaxy Plus) [26] by Nenad Dunjić and Milan Tadić. This project was also financed by the "Institute for learning and teaching resources". Galaksija Plus was actually the original computer Galaksija, with 48 KB of RAM, the high-resolution graphics and the three-channel sound generator.

6.2. Lola 8

One of the rare home computers that reached mass production was the Lola 8a, developed as a laboratory project of the IM "Ivo Lola Ribar". The team from their laboratory has worked on the development of hardware for industrial controllers (the project PA 512) built into the machines that "Lola" successfully exported at the time. Their knowledge was applied to a small computer Lola 8. The hardware designer was Radovan Novaković, and the software was written by Nela Radovanović [27]. The project proceeded during 1983 and 1984, so Lola 8 in various stages of development was shown on

local fairs - there was a joke saying that "Lola computer is developed from fair to fair." Visitors particularly remember the games in which, if you collected enough points, you could enjoy the sound of a song "Fijaker stari" (The Old Carriage).

Fig. 6. Lola 8a, project of the IM "Ivo Lola Ribar"

During the summer of 1984, the "final" version of the computer was presented to the public. It was based on Intel 8085 processor with 16 KB ROM and 6 KB of RAM used to store user programs written in BASIC or machine language. The keyboard was non-standard, and 48 keys were not positioned in the usual, "staggered" manner, but one above another, resembling the industrial controllers. The computer had a sound generator with three independent programmable channels, and it used the tape recorder as the external memory. The price of 55,000 dinars was announced, which in mid-1984 was equivalent to 600 Deutschmarks.

Production of the Lola 8a actually started much later - on May 25, 1985. The first "Youth Computer Factory" was opened in Belgrade [28] and it employed thirty workers who had been making Lola 8a. It was substantially revised version of the original Lola 8, with the classic QWERTY keyboard, an Intel 8085 microprocessor working on 4.9 MHz, 24 KB ROM and 16 or 32 KB of RAM. The graphics resolution was 320 * 300 points, and the software was considerably improved, which made the Lola 8a software incompatible with the original model.

During its production cycle, about 2000 units of Lola 8a were made and delivered to primary schools in Belgrade, secondary military schools, and military academies. Publishing company "Nolit" tried to offer the computer on the market, at the price of 125,000 dinars (about 980 Deutschmarks). In the

"Youth Computer Factory" the management tried to organize the team to develop software for the Lola 8a, but the overall software support for this computer was modest, resulting in the decrease of interest and the market for it in the following years.

6.3. TIM computers

During the seventies and the early eighties, the engineers at the Institute Mihajlo Pupin designed a number of control and measuring devices, peripheral units, industrial controllers and robots. In 1985 the Institute started manufacturing the equipment for PTT Serbia. The result was a specialized computer TIM-001, and later the TIM-100. A large number of these computers have worked at the post office counters.

The computer offered on the market by the Institute Mihajlo Pupin, TIM 011 [30], has emerged through an alternative channel, sharing only the computer case and the name with the TIM family. In 1987 the Pupin Institute bought the project of CP/M computer that was developed by Nenad Dunjić, Milan Tadić and Ljubiša Gavrilović. The computer was offered to schools, but also in a build-it-yourself campaign, supported by the special edition of the Computers magazine, under the title "Computers in Your School" (April 1988) [31]. Ambition was to repeat the success of the build-it-yourself campaign of the Galaksija computer, and the results were satisfactory, but far from the original.

The TIM 011 was based on the microprocessor Hitachi HD-64180, and the operating system ZCPR 3. However, things were not quite as unusual as these names suggest. The HD 64180 is a hardware and software upgrade of the Zilog Z-80, which, with the built-in MMU, had the ability to address 512 KB or 1 MB of memory divided into 4 KB pages. The speed-up of the processor was based on the two-channel DMA controller built into it, and the construction of the computer was simplified by introducing the embedded interrupt controller, two serial interfaces, and two counters.

In its basic version, the TIM 011 had 256 kilobytes of RAM and additional 32 KB of video memory, located in the I/O map of the microprocessor. In the text mode, it could write 80 characters in each of the 24 rows on the screen, and graphics resolution was 512 * 256 points with four levels of gray. As an external memory, it used the double-sided 80-track floppy 3.5" disk drive, which provided the capacity of 780 kilobytes [30].

The operating System ZCPR 3 (more precisely, the operating system is Z, while ZCPR 3 stands for operating system with command processor - Z80 Command Processor Replacement) is a variation of the CP/M 2.2. ZCPR 3 is an open source operating system, which allowed the authors of the computer to implement the necessary changes. Microsoft CP/M BASIC 5.21 was distributed together with it, and it was extended with instructions that supported graphics and sound. The TIM 011 was a rather fast computer - the PCW speed tests put it at the top of the list, along with then-champion in speed, Acorn BBC B computer [30]. It was an interesting concept that

reached the limits of the 8-bit technology. It was distributed to schools, the attempt was made to sell it on the market and organize the build-it-yourself campaign, but the results were modest, as the companies and individual buyers have already turned to IBM PC compatible computers.

The final representative of the TIM computer family was the TIM 600, introduced in 1988 as the first Yugoslav 32-bit super-microcomputer system. The TIM 600 was based on the (then) top-of-the-line Intel's processor 80386 (with an arithmetic coprocessor Intel 80387), and the original hardware that relied on three system buses: the 32-bit bus between the CPU and the memory, the 16-bit bus for I/O operations and the 8-bit bus for SCSI devices. Unix System V.3 was chosen for the operating system.

Ambitious promotion of the computer TIM 600 at the 32nd International Technical Fair in Belgrade (May 16-20, 1988) included the presentation of the first Serbian aircraft SARIĆ, that flew in 1909, only six years after the Wright Brothers' historic flight (1903), and the part of the CER-10 computer, which was constructed 12 years after the ENIAC. The TIM 600 has continued the tradition, having been promoted barely a year after the first of IBM's 80386 computers [32].

The promotion has attracted considerable media attention, satisfying the expectations of the CEO of the Institute Mihajlo Pupin, Draško Milićević [33], whose ambition was to create the Technology Park at Zvezdara, and stop the brain drain of young and talented engineers. Unfortunately, although the TIM 600 was a fascinating engineering project, it was designed without considering the needs of the market. For the sake of some technologically superior solutions, its designers have given up the PC compatibility, so it became clear that the computer had no chance of commercial success. However, it deserves to be mentioned as one of the few attempts of keeping up with technologically developed countries.

6.4. Pecom 32/64

Elektronska industrija Niš (Electronics Industry Niš) has joined the competition for the school computer at the end of 1985, promoting the Pecom 32, and later the Pecom 64. These computers were looking unusual - rather compact, with the built-in power adapter and the keyboard based on QWERTY scheme, but without special characters and the Serbian letters, and with an unusually short spacebar [34], Figure 7. The CPU was the CDP 1802 working on 2.8 MHz - an advantage of this rather old processor was considerable robustness to low temperatures and other extreme conditions. It was the first microprocessor in space, used in the Voyager mission - one of these processors was used in the craft that leaves the Solar system these days. However, this feature of the Pecom was not overly important for a home computer.

The Pecom was functionally similar to a home computer COMX-35 [35] which was sold in the Far East by the company COMX World Operations, while in Scandinavian countries it was distributed by West Electronics. There

were a lot of (mostly free) games for COMX-35, and some of them could work on Pecom computers.

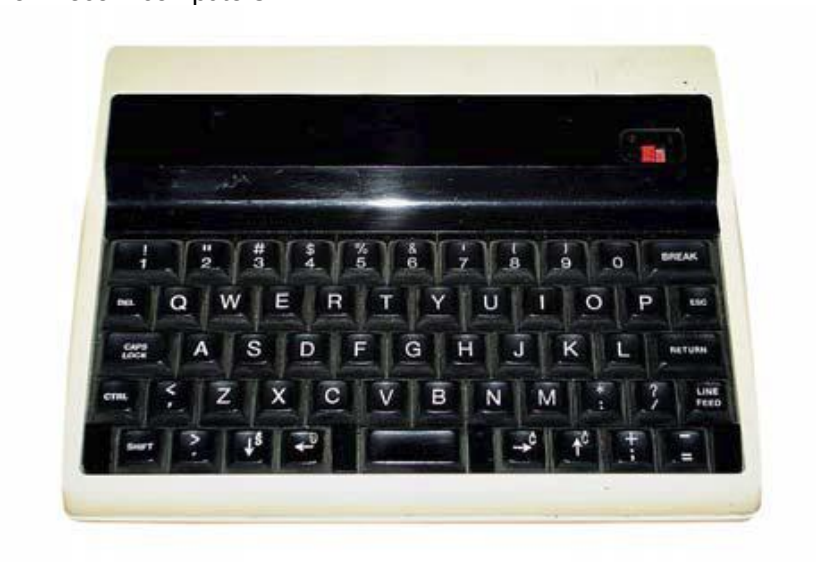


Fig. 7. Pecom computer

The Pecom had 32 KB of RAM and 16 KB of ROM. Text could be displayed on TV set using 40 columns * 24 rows of characters, from a character set that could be redesigned in software. This feature could be used for graphics emulation, so a skilled programmer could achieve the resolution of 240 * 216 points in eight colors. The Pecom had a basic sound generator, controlled from BASIC or the machine language program. The BASIC interpreter was unusual, with lots of interesting new commands and the possibility of pseudo-compilation: using the command RUN+, it was possible to store the absolute destination address with each GOTO and GOSUB instruction, thus speeding up the following program executions. Nevertheless, the Pecom was not the speed champion, and work with it was made difficult as the error messages were represented in the form of codes, so the user could not avoid frequent browsing of the user manual [34].

The Pecom computer reached the stage of serial production. It was delivered to schools, and sold on the market, mostly in the bookstores. The production run was not published, but Pecom computers could be seen in the windows of the bookstores a long time after the production cycle was completed. Efforts were made to develop and offer the educational software for it, but the software support remained modest.

6.5. The choice of school computer

During the mid-1980s it became clear that the computer education was becoming "a new literacy", with the necessity to be seriously studied in the elementary schools. The inevitable consequence was the need to start the first round of the domestic computer industry production. Based on the number of existing schools in the country, and the former centralized decision-making, the producer of the future school computer could count on the reliable market for the large series of machines sold.

From the first days it was clear that the process was complex. The Education Council could not decide between the Galaksija and the Lola 8a, so they found a Solomonic solution to buy both computers for each school in Belgrade. Unfortunately, these computers were completely incompatible, and the development of the educational software for both of them was irrational and expensive.

In the late 1980s it became clear that the school computer could only be a PC clone, but the domestic production of such computers was complicated and expensive, and the authorities constantly insisted that the school computer should be a local product. Thus, in different regions and different times, schools have purchased Galaksija Plus, Pecom, TIM 011, Oric Nova (after the disappearance of computer Oric-1 and Oric Atmos on the British market, the license was purchased through the Slovenian Avtotehna, and the production was initiated in Serbia, so it also become a local computer), EI Lira (interesting PC XT clone), TRS 80 Color Computer, etc.

Even when the computer was selected, its production in larger series has been a serious challenge. Poor supply of components, problems with import, the lack of quality control and related problems have made computers arrive to school in semi-working conditions, so its exploitation depended primarily on the enthusiasm and knowledge of the teachers. The teachers were often not skilled enough, so they felt uncomfortable in front of the students, many of whom had better computers at home, so they already knew a lot about them. That is why many school computers remained unused.

6.6. PC years

The early production of PC computers was present during eighties in some companies, like NOVKABEL from Novi Sad, which produced ERA family of personal computers from 1979 till 1987. However, serious expansion of the domestic computer market began in 1990, with the establishment of numerous private companies, the introduction of convertible dinars and removal of the import restrictions. PC clones were first imported from Austria and Germany, and then from the Far East, with a steady fall in prices under the pressure of strong competition. In the years of economic sanctions, the computer market was still there, with constant problems with the import, so it turned to the cheapest (and therefore low quality) components. Since the

import of the assembled computers was complicated, the companies started domestic manufacturing of PCs with the components purchased from various sources. Each delivery of components dictated a new configuration of the products, various customers got different models, so the maintenance was difficult, especially with a mass distribution of illegal software.

Expansions of the market lead to the formation of the first brands and the introduction of mass production, with the quality control. Some of the top brands were created by Comtrade, Jugodata, and Pakom, which standardized the equipment to be delivered. After 2000, the foreign brands reached Serbian market, but there are still many successful models that are constructed, configured and assembled in Serbia.

7. Conclusion

Although the population and the computer market in Serbia are not comparable in size with the population and the markets of the developed countries, computer boom arrived everywhere in a similar way. From the first computers composed of vacuum tubes, inspired by the need to decipher codes, which were huge in size and hard to program, to the first home computers offered in a kit by popular technical magazines, the developments in the computer history of Serbia resembled that of the technological leaders, with several years of delay. It had its own inventors and its own heroes, just like the computer pioneers that Steven Levy called "The Wizards and Their Machines" [14]. Home computers started with several mutually incompatible models that competed for the favor of users. Prohibition on imports of computers, lack of resources and the inert large economic systems in state-ownership has led to significant delays in the introduction of modern technologies, which is still visible nowadays. However, the research and development in the area of computer science and computer engineering, which lead to the production of original computers in Serbia, also influenced the higher education in this area [37], which is nowadays integrated to the European higher education area, using best practices on the bachelor and master level [36]. The history of the last 50 years of digital computing shows that the potential of human talents and creativity can lead to original and competitive solutions in the area of computing in Serbia.

Acknowledgements. We are indebted to Professor Dragana Bečejski Vujaklija and Nikola Marković, the President of the Informatics Society of Serbia, for inviting us to contribute to the publication that they prepared in honor of the celebration of the first 50 years of digital computing in Serbia, which also inspired us to write this paper. We are also heartily thankful to Professor dr Borivoj Lazić and Professor dr Jovan Đorđević for providing valuable information and support.

References

1. ENIAC Museum Online, Available: http://www.seas.upenn.edu/about-seas/eniac/ (current May 2011)
2. Dan Lynch, Bert Herzog: A Time for Celebration. Communications of the ACM, Vol. 38, No. 12, 11-12. (December 1995)
3. Nikola Marković: Fifty Years of Digital Computing in Serbia - Catalogue, Hall of the National Bank of Serbia, (December 2010)
4. Djordje Paunovic: On the 100th Anniversary of Lecturing Electrical Engineering in Belgrade, YUMTT Chapter Informer, 1-3, (October 1995).
5. Forty Years of Computers in Serbian Mining, [Online]. Available: http://www.fmgseca.net/izlozba/ (current January 2011)
6. Dušan Hristović: CER-10 – The First Digital Electronic Computer in Serbia, IT Star NL, v7p6-7, (Spring 2009).
7. Our 50 years, publication of the School of Electrical Engineering, Belgrade on its 50th anniversary (2003)
8. Dejan Popovic: Rajko Tomović - Scientist, Teacher, and Friend (1919- 2001), Journal of Automatic Control, University of Belgrade, Vol. 12, 1-3 (2002)
9. Rajko Tomovic, Ahmed Mandzic, Tihomir Aleksic, Petar Vrbavac, Vukasin Masnikosa, Dusan Hristovic, Milojko Maric: Digital Electronic Computer CER-10 in the Institute BK Vinca, in Proceedings of the 5th ETAN Conference, Belgrade, pages 305-314 (November 1960)
10. Vladan Batanovic, Dusan Hristovic, Cedomir Milenkovic: Development in the Computer Field in the Mihajlo Pupin Institute, *Info m,* 36/2010, on CD (November 2010) , in Serbian
11. Vladislav Paunovic, Dusan Hristovic: Review and Analysis of the CER Computers, in Proceedings of the 44th ETAN Conference, Sokobanja, pages 79-82 (June 2000)
12. Stanko Stojiljkovic: Successful on a Secret Mission, Politika, (September 19, 2006), in Serbian
13. Paul E. Ceruzzi: A History of Modern Computing, The MIT Press, second edition (2003)
14. Steven Levy: Hackers, Heroes of the Computer Revolution, O'Reilly (2010)
15. Computers in your home 2, BIGZ, Belgrade, (July 1984), in Serbian
16. Dejan Ristanović: Galaxy at School, in Computers in Your Home, page 66, Belgrade, (December 1983), in Serbian
17. Voja Antonić: Galaxy Computer, in Computers in your home, 50-64, Belgrade, (December 1983), in Serbian
18. Voja Antonić: A Thousand Whys, Galaxy No.140, page 31, Belgrade, (December 1983), in Serbian
19. Dejan Ristanović: Galaxy Computer - The User Manual, Institute for learning and teaching resources, Belgrade, (1984), in Serbian
20. Dejan Ristanović: Computers in your home, BIGZ, Belgrade, (December 1983), in Serbian
21. Dejan Ristanović: Galaxy Computer, Galaxy No. 138-140, Belgrade, (October 1983), in Serbian
22. Jovan Regasek: Memory for Galaxy, Computers in your home 2, BIGZ, page 88-90, Belgrade, (July 1984), in Serbian
23. Nenad Dunjić: Fine Hardware for Fine Graphics, Computers No. 6, page 63-67, Belgrade, (June-July 1985), in Serbian

24. Dejan Ristanović: Programs Broadcasting, Galaxy No. 143, page 64, Belgrade, (March 1984), in Serbian
25. Dejan Ristanović: ROM 2 for Galaxy, Computers in Your Home No. 3, Belgrade, (December 1984), in Serbian.
26. Dejan Ristanović: Galaxy Plus, Computers No. 8, 6-8, Belgrade, (October 1985), in Serbian.
27. Jelena Rupnik: The Woman Who Created LOLA, Galaxy No. 148, page 59, Belgrade, (August 1984), in Serbian.
28. Jelena Rupnik: How they Make LOLA, Computers No. 10, page 18-19, Belgrade, (December 1985), in Serbian.
29. Bogdan Janković: CER of our Youth, in Computers that we Loved, PC Extra, PC Press, (January 1996)
30. Dejan Ristanović: TIM is Better, Computers No. 35, page 7-9, Belgrade, (February 1988), in Serbian.
31. Computers in your School, BIGZ, Belgrade, (April 1988), in Serbian.
32. TIM Bridges the Gap, Computers No. 39, page 3, Belgrade, (June 1988)
33. Vesna Ćosić: Draško Milićević: My Goal is Profit, My Tool is Science, Computers No. 48, page 6, Belgrade, (March 1989), in Serbian,
34. Nenad Balint: Pecom, Computer World, page 14, Belgrade, (May 1986), in Serbian
35. Comx-35, Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/Comx-35 (current April 2011)
36. Klaus Bothe, Zoran Budimac, Rebeca Cortazar, Mirjana Ivanović, and Hussein Zedan: Development of a Modern Curriculum in Software Engineering at Master Level across Countries, ComSIS, Volume 06 , Issue 01, page 1-21 (June 2009)
37. Jelica Protić, Borivoj Lazić and Jovan Đorđević: The Beginning of Digital Computing in Serbia and the Role of the School of Electrical Engineering in the Development of Scientific and Educational Work, *Info m,* 36/2010, on CD (November 2010), in Serbian
38. Dejan Ristanović: Home Computers Made in Serbia, *Info m,* 36/2010, on CD (November 2010), in Serbian

Jelica Protić is an associate professor at the Computer Engineering and Informatics department at the School of Electrical Engineering, University of Belgrade. She holds a Ph.D. (1999), M.Sc. (1994) and B.Sc. (1987) degrees in Electrical Engineering and Computing, from the same University. Her research interests are in programming languages, distributed systems, computer networks, and all aspects of computer-based quantitative performance analysis and modeling. As a Vice Dean of Tuition from 2004 until 2009, she was involved in the accreditation process, as well as implementation of computer technologies in teaching and learning, including the development of dedicated software tools to support these processes. She coauthored the book *Distributed Shared Memory: Concepts and Systems*, published by IEEE CS Press, as well as over 80 papers published in scientific journals and conference proceedings.

Dejan Ristanović is the founder and Editor-in-Chief of the computer magazine "PC Press", and the cofounder of SezamPro, one of the leading Internet providers in Serbia. He is the author of the first publication about personal computers in former Yugoslavia, published in December 1983, and cofounder of Sezam, the biggest BBS in former Yugoslavia. He is the author of 18 books, 15 CD/DVD publications, and more than 1500 articles published over the 30-year span.

Received: May 05, 2011; Accepted: June 13, 2011.

DOI:10.2298/CSIS100326001L

Workflow Specification for Interaction Management between Experts in a Cooperative Remote Diagnosis Process

M.A. Laredj and K. Bouamrane

Department of Computer Sciences, Sciences Faculty
University of Oran - (Es-Senia)- Algeria
BP 1524, El M'Naouer, 31000
{lmadnane, kbouamranedz}@yahoo.fr

Abstract. The maintenance became one of the strategic industrial functions. Far from being stabilized today, it progress by introducing the new management methods and the technological development of the production equipments, in particular in the measurement and the control operation. All these factors modify the organization modes of the maintenance function, which converges more and more towards a cooperative approach. It will absorb a big part henceforth, and subsequently, will require a rigorous modeling allowing its future implementation. In this fact, we propose the specification of a workflow model to assist the cooperative maintenance. It main aim, will be to coordinate the interactions between the various intervening actors in the maintenance process.

Key words: E-maintenance, CSCW, Groupware, Workflow, OSSAD, Petri nets.

1. Introduction

Maintenance is today, an activity where collaboration is fundamental. Many actors collaborate to achieve a common purpose: reducing the failure probability or object degradation, each one, having its personal knowledge and practices. Thus, several people of diverse countries and professions, working for distinct companies, can have to work together for the product maintenance, which implies that various cultures must be taken into account in the management of the maintenance process. This is not without causing many problems. It is then necessary to reinvent the organization of the maintenance process and to redefine the relationship between the experts within the cooperative work. The roles and the positions of each actor must be redefined.

In this work we will define in first section the various forms of maintenance in order to put forward in the second part the importance of the cooperation between experts during failure diagnostic. After the description of the various

research and platforms, allowing the implementation of a cooperative and collaborative maintenance vision. We will focus ourselves on Proteus platforms [2] and the contribution of Saint Voirin [3][4] which we will improve with “Workflow” approach. We will present in following section the workflow modeling through two steps. The first one based on a systemic language “OSSAD” will be formalized at the second step by the Petri nets that allowing the simulation and validation of our Workflow models thanks to the mathematical properties of the Petri nets.

2. Remote maintenance and diagnosis

It consists to maintain a functional unit, ensured by Internet or a direct telecommunication between this unit and a specialized center. It’s characterized by:

- A remote service to support the diagnosis and repair.
- An expert system to support the failure diagnosis.

The alarms interpretation during monitoring phase can be divided in three parts:

- The filtering limits the alarms information load, and tries to present only “interesting” ones.
- The localization characterizes or identifies the detected dysfunction situation.
- The diagnosis proposes the most credible sources of the observed dysfunction. We often reserve the term of breakdown to the result of the diagnosis. The purpose of the diagnosis phase is to seek the main causes of the observed phenomena. It is thus about a major analysis of the process. The remote diagnosis requires knowing the most possible information on the remote system.

Remote maintenance and e-maintenance can be expensive in terms of costs and quality in some hardware configurations and application fields. In industry, it applies to systems (machines, automats…), connected by Internet or a communication network to the maintenance centers. In the case of systems failures, the maintenance center is automatically informed and can start remote operations. Consequently, we avoid expensive displacements of the supplier specialized technicians for a few minutes of intervention. They allow also rapidity and efficiency of intervention to answer any specific request and in some case, ensures the safety of the expert in dangerous operations as on high lines voltage, or in the nuclear industry.

3. Related works in e-diagnosis and e-maintenance platform

A recent literature review related to this topic with emphasis on Web technology and multi-agent systems has been presented by Campos [19]. Campos et al. concluded that the current developments in these areas are still at the rudimentary stage. For Jardine et al. [20], the reasons that e-maintenance technologies have not been well implemented in industry are :

- lack of data due to incorrect data collecting approach, or even no data collection and/or data storage at all;
- lack of efficient communication between theory developers and practitioners in the area of reliability and maintenance;
- lack of efficient validation approaches; and
- difficulty of implementation due to frequent change of design, technologies, business policies, and management executives

However, there exist considerable incentives in developing tools, methods, or systems for solving these issues, and several e-maintenance platforms have been developed and are in use today, these platforms are a result either of the industrial world or of the academic one. Muller et Al [16] classified them as proprietary platforms (i.e. ICAS [18]), platforms for research and education (i.e. TELMA [1]) or platforms developed within projects (i.e. PROTEUS [2]).

- Muller et Al [16] conclude that among the future common industrial/academic working/research directions, several can be underlined:
- Modelling and implementation of the new processes (e-monitoring, e-prognosis, e-logistics, etc.).
- Need of theory and tools for mastering the behaviour of the interactions of the system–maintenance–economy model, and maintenance decision support system for cost-effective decisions.
- Development of new infotronics-based e-maintenance systems integrating new protocols for collaboration and negotiation, maintenance **workflow**, maintenance Web services, etc.

The PROTEUS platform attempted to correct this needs; it brings a contribution of the vertical integration of applications in the domain of remote maintenance of industrial installations [2]. It provides a unique and coherent description of the equipment (through an ontology description), a generic architecture (based on the **''Web services''** technology) and coherent models of heterogeneous components. However, even if these approaches make it possible to improve the speed and the reliability of the maintenance actions, in fact, it doesn't exploit (or incompletely) cooperation and collaboration between experts, whereas this one can be a decisive element in breakdowns detection and diagnoses. Since these platforms often insist on the system

interoperability (GMAO, SCADA…) or the communication between the physical sensors by forsaking the human aspect which is summarized in the form of a pseudo GMAO through request for control generated automatically using an expert system more adapted to the process of preventive maintenance than corrective one.

Saint-Voirin [4] tried to optimize the cooperation activity on the PROTEUS platform. Actually, they have implemented their own conceptual cooperation meta-model on the PROTEUS platform in order to simplify design operations. Their meta-model builds on the use of **multi-agent** systems allowed computer models and simulation of the remote maintenance cooperative system. He called his approach as **Scoop methodology** [3].

3.1. *Scoop methodology*

This methodology helps to draw structural models of the system. These models are easy to read and their graphical aspects simplify comprehension of the system. The structural modelling is based on a nomenclature of members and interactions. Human members are represented using a square containing basic information. Equipment members are represented using a circle containing the basic information. Interactions are represented using specific nomenclatures, the communication nomenclature and the shared data access nomenclature. In both interaction nomenclatures, the square symbol means that mutual exclusion is required for this type of interaction. The number of arrows is related to the number of members involved in the interaction.

However this nomenclature describes the global structure of the system. It is very useful for specification. However, we cannot verify or simulate anything with this representation. That is why Saint-Voirin [4] developed associated Petri nets for the interactions. Petri nets created are used to verify livingness properties, to find deadlocks and conflicts in the interaction. He also developed an analysis based on stochastic simulations of these Petri nets [4]. Nevertheless, interaction study is just a part of cooperation aspects in cooperative systems. Human cognitive aspects have to be represented. To study this particular point, Saint-Voirin choose to create a multi-agent simulator of cooperative systems. He proposes to define each member of cooperation as an agent. Multi-agent systems allow him to propose a good representation of human entities because of their artificial intelligence abilities. Finally, Saint-Voirin used a XML formalism that allowing description of simple knowledge classified in the identified parts: skills, environment and role and goals.

3.2. Contribution

Saint voirin used in Scoop methodology [3][4] various models (Petri Net, UML, Stochastique Petri net PLOOM-UNITY, Multi-agent, XML…) through several phases (Formal specification, Structural modelling, Interactions modelling, Behavior and knowledge modelling) to define the cooperation and coordination between experts on diagnosis process. In addition he created his own nomenclature that is not well known, so we will try to simplify his approach by using workflow with well know modelling language. This language must be able to generated automatically Petri without human assistance (maintenance experts or technicians) for validation and simulation.

We'll aim to develop an application which assists the cooperative remote maintenance based on workflow architecture **fig (1),** it will build an operational and autonomous system, whose main aim is to coordinate the interactions between the various intervening actors within the maintenance process.

3.3. From Groupware to Workflow

The classification of J. Grudin [7] identified a particular type of groupware, dedicated to the management process (industrial, commercial, administrative, etc) and to the coordination of the various intervening during this same process. This particular type of groupware Known as **"Workflow**" takes care of the good circulation of the documents and information between the various intervening at the key times of a cooperative process such as cooperative maintenance. This is why we choose to use them for implementing our system. We propose a ***Workflow*** system who automates the management and the coordination of the information flow following pre-established models.

The tasks of data processing pass from a person to another according to a well defined conditional circuit. Each actor (technician, expert…) of the circuit carries out its task without needing to be concerned with what was made before and of what should be made afterwards. The application presents to the user the necessary information to carry out his task, before the process does not follow its course towards the next step when the individual makes the task

The possibilities offered by the workflow tools are:

- *Rigorous regulation of the procedures*: the regulation of the task sequencing guarantees the execution of a business in accordance with the workplan.
- *Flow Control of the work*: The workflow software makes it possible to follow the progress report of a business step by step and to detect quickly possible bottlenecks corresponding to the accumulation of works at a station.
- M*aximum of automation*: waste of time due to move, seek, photocopy, distribute and classify the documents are decreased considerably

- The workflow software offers also the possibility to automate all the operations for which a human intervention does not bring a real added-value.

3.4. Workflow modeling

The specification of the Workflow applications involves describing precisely in models form, the actors implied in the realization of a cooperative task, the interactions structure that link this actors, the information nature exchanged and the dynamics of the treatments which must be carried out. However, each year, tens of workflow are specified for several companies. In the best cases, the development team bases itself on a rigorous method of specification resulting from the Software engineering. But very often, it bases on a "home made" method resulting from an adaptation of an old method (such as SADT for example) [8].

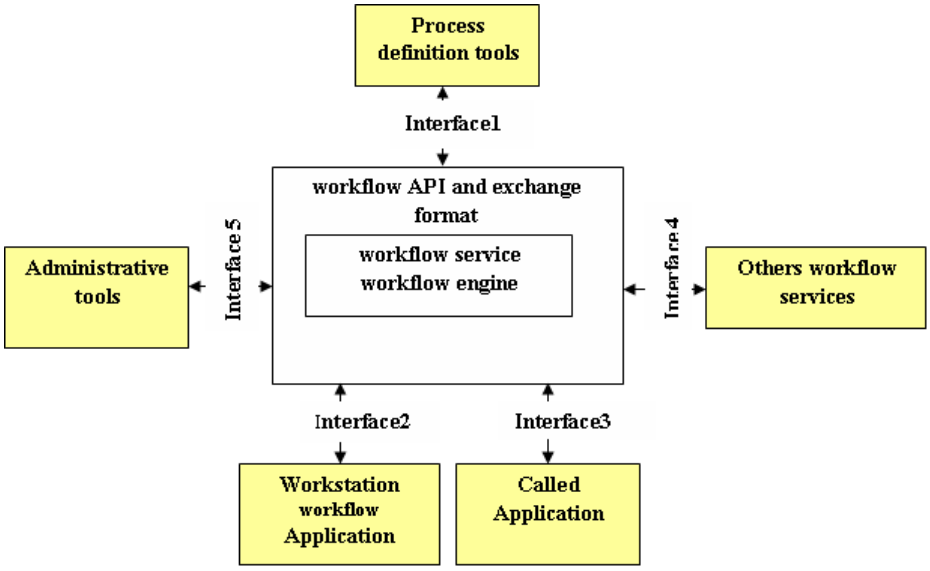


Fig 1. Workflow reference Model **[WfMC]**

It is then frequent to note that the developed interactive systems pose many problems and do not always meet the user's needs, and are often badly adapted to the work organization [8] [9]. This is due to an inadequacy between the methods used and the target aim. The necessity to adapt the methods due to the fact that no uniform method of workflow modeling and specification. The developers which feel a lack during the application of "their" method to a new situation try to improve it, according to their own criteria. It engenders an expansion of personal methods often missing coherence on some aspects.

3.5. Which language for the workflow modeling?

To conclude the modeling and the specification of the remote diagnostic process, it is necessary for us to find the best work organization that permit to provide to each actor the technological tools which assist or automate its individual work and in same time enabling him to communicate with the others in order to coordinate the various activities and thus to achieve the common goals [10]. A complete method would have to:

- Be sufficiently general to allow to model any business process (even if it comprises phases which cannot a priori be implemented by a workflow).
- support the analysis since the identification process until the modeling procedures which will automate the flow .
- Reasoning about the goals and not on the functions performed by different service organization.
- Enable organizations to address complex processes that are not clearly defined. A systemic approach is required in this case [10].

The comparative study of modeling methods, from software engineering (Merise, SADT, SART, OMT, OOM... etc) [10], allowed us to conclude that they are all oriented towards the structuring of data and automated processing, neglecting organizational aspects. This leads us to push our investigations into methods used less or more recent and longer correspond to our expectations. Thus, we have discovered the method OSSAD [11] which is oriented towards the organization of men's work rather than to the organization of data and automation of treatment.

3.6. The OSSAD Method

The OSSAD Method [11] (Office Support Systems Analysis and Design) was developed during the ESPRIT program (European Strategic Program for Research in Technology Information) from 1985 to 1990 by a multinational team of consultants, academics and Technology Information users. It is about a systemic approach which helps to understand how people work together, by including the users in the conceived system. OSSAD is thus interested above all in organizational operation. It is a method which makes it possible to analyze how various people coordinate their tasks in order to provide a global result. It aims to:

- Provide stakeholders a conceptual framework and organization of work to enable them to lead a project.
- Allow the adaptation of the general framework to each specific situation.
- Provide modeling tools of tertiary or administrative work
- Allow interaction design (and not separately) the sub-technical systems and human

- Propose new opportunities for dialogue between managers, technicians, users of technology.

This method proposes an approach which is done in three stages. Three different levels are thus established: *abstracted*, *descriptive* and *prescriptive*. They meet all needs clearly defined:

Table1. Levels of OSSAD modeling

LEVEL	ROLE	PURPOSE
Abstract	Purpose modelling	What we have to do or to reach?
Descriptive	Ressources modelling	How we realize the purposes? Whith what and who?
Prescriptive	Workflow specification	How did automate the ressources?

4. OSSAD cooperation modeling within a remote diagnosis process

Our modeling is based on a cooperation management algorithm of an expert group suggested by Boussedjra [12] to establish diagnoses and the maintenance of the detected breakdowns. The algorithm manages the group organization, and the communication between experts, it is based for that on the following assumptions:

1. Each grouped together for treatment failure declared by a technician is a group.
2. At any moment, one group member diffuses its data and all other members shall be on standby.
3. Experts are multipurpose or general (they do not know the installations)
4. The site may not be cooperating in the delivery status unless it has been authorized by the coordinator.
5. At any moment one and only one person is authorized to speak or diffusing data.
6. ***The creation of a group*** is initiated by the technician, Fig (2).
7. ***The attribution of sequence number*** is made according to the arrival time of the reply messages. The built group is composed of two sub-groups:
   - The first contains the cooperating experts for the resolution of the breakdown and a coordinator. It under group is active: Exchange of information between the members and a coordinator.

- The second under group is consisted of the members of the active group and the technician (it under group is optional).

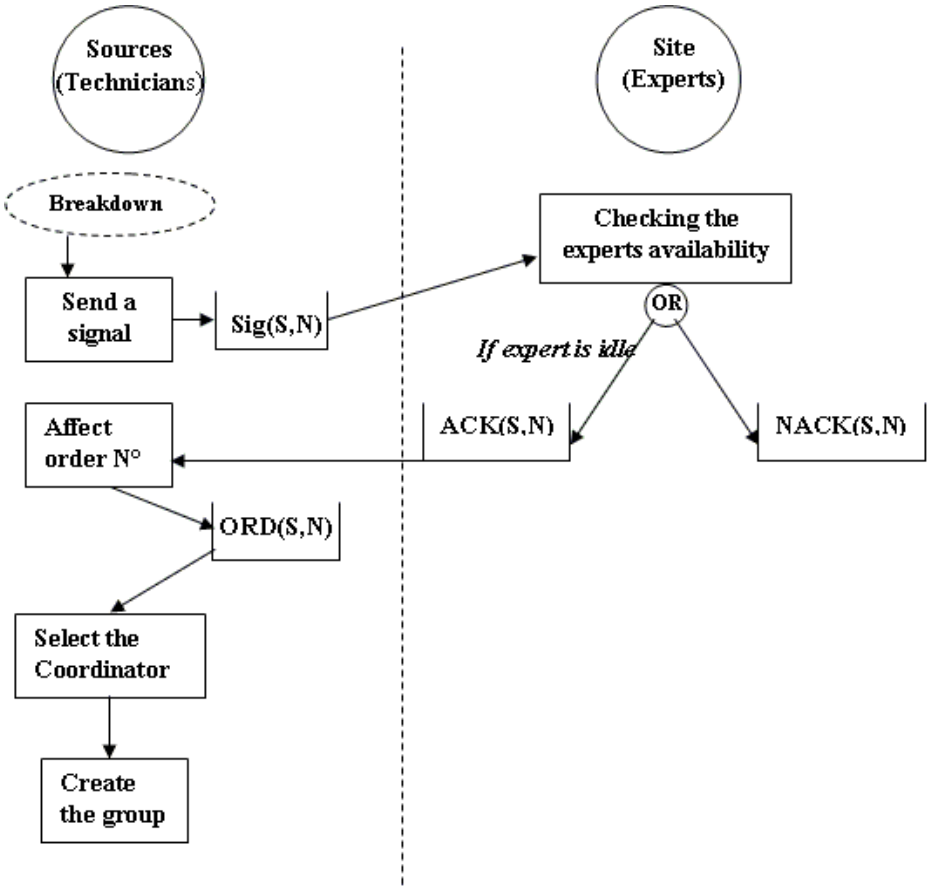


Fig 2. OSSAD operation Model of an expert group building process

8. The group coordinator selection according to the quality of the network, between him and the breakdown site. The coordinator's role is to act as an interface for communication between group members and the outside world (site failures or other cooperating groups).

9. ***The dissolution of expert group*** can occur in order to answer the set of the breakdowns declared. An expert group can be built by assigning free experts to the declared breakdown.

10. the ***treatment of a new breakdown*** B if all the experts are occupied on a breakdown, will be as follows:

11. If the treatment of a breakdown in progress is completed, then the new breakdown B is treated immediately and the group is rebuild.

12. If the treatment is not finished yet, but an assignment of one or more experts to the declared breakdown B is possible, then two new groups are built, one for the treatment of the breakdown B another for the treatment of the breakdown A.
13. Otherwise, if the declared breakdown B cannot be treated, then it is stored in a queue like a future work.
14. ***The assignment for a new group of expert*** can be done according to the experts sequence numbers (while assigning to the new existing group) or per decision of the each group coordinator.
15. ***The addition of a member*** is done by a call or an invitation of the group via its coordinator, or then by a request from a free site wanting to join the group. As long as the two sites are not agreement (reception of acknowledgment of positive delivery), the member does not enter in the group, fig (3).
16. ***The management of mutual exclusion*** is taken into account thanks to the requests for authorizations managed by the coordinators and the cooperators sequence numbers. The requests classified by importance are sometimes inserted in queues, fig (4).

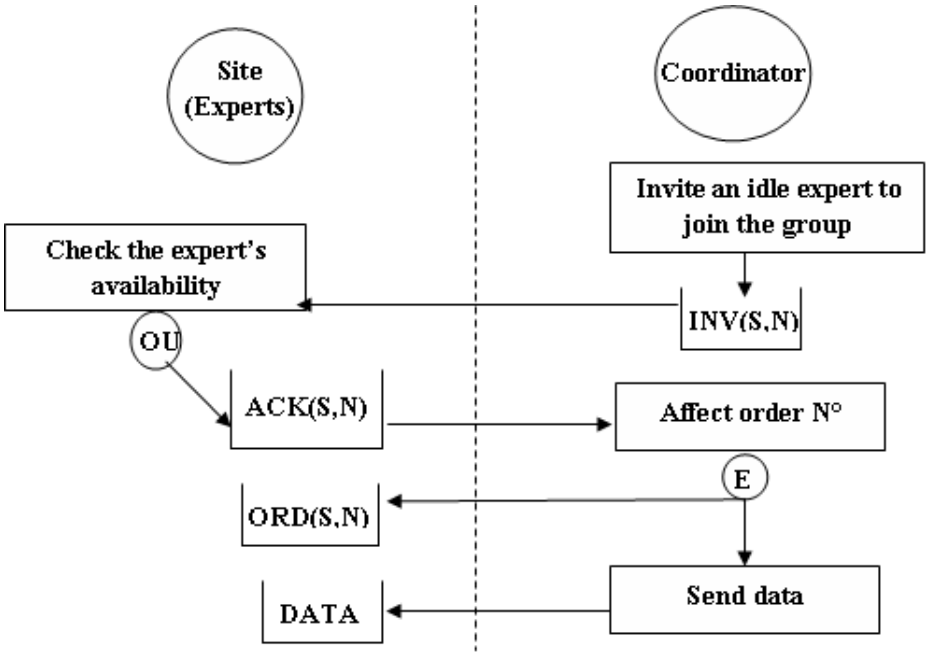


Fig 3. OSSAD operation model of the new member adding process by invitation

4.1. OSSAD prescriptive Model (workflow)

The OSSAD descriptive operation models presented below (*fig 2, 3, 4, 5*) do not constitute (in this form) a specification allowing the workflow applications generation. Chappelet and Legrand [11] introduced a additional *prescriptive* level into OSSAD, this last extends the operations model by the specification of what will be automated in a workflow. This is summarized in the concepts of: *Document*, *State* of document, *Structure* of document, *constraint* of prohibition or obligation, *Completion date* of an operation, *Selection* and *Notification*.

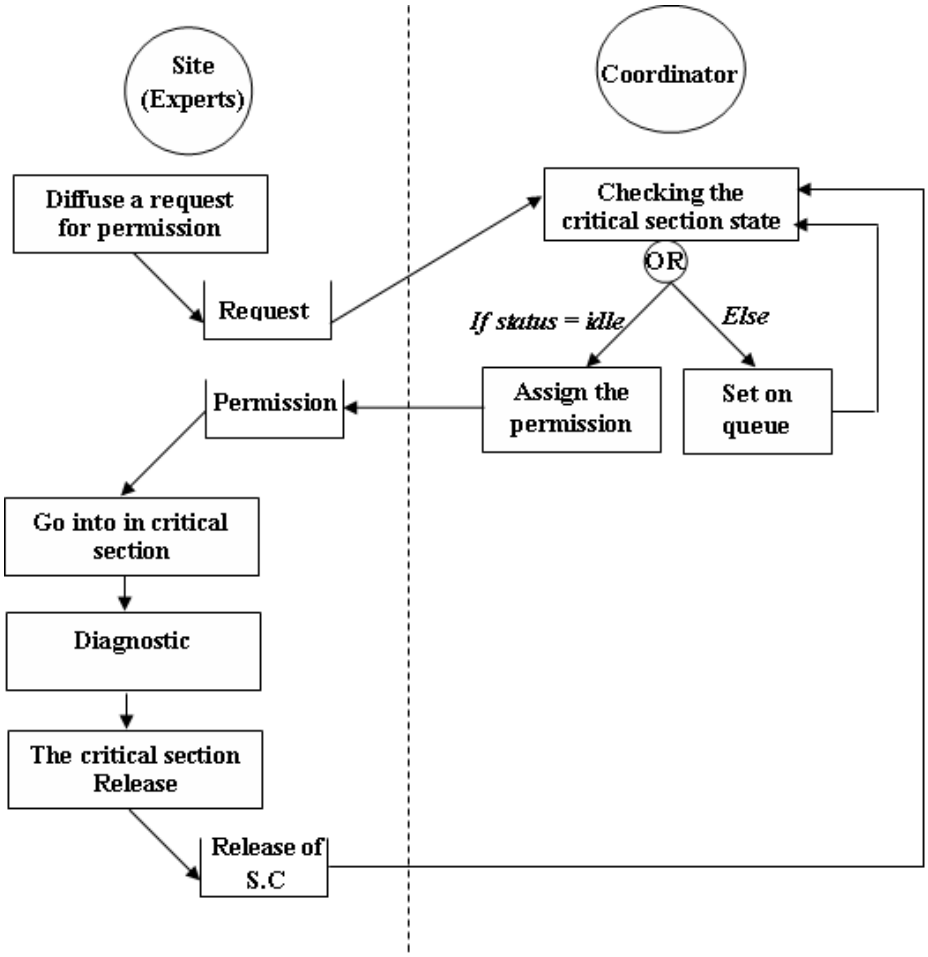


Fig 4. OSSAD operation Model of the mutual Exclusion process

The transformation of a descriptive model into a prescriptive model is done according to the following steps:

- Identify the resources which will be computerized. These resources become documents.
- Specify state changes of these documents in the operating flows (including changes calculated by the workflow management system)
- specify constraints between operations, if necessary,
- Determine the states where it is necessary to select the actor or actors to perform the following operation. This selection can be associated to a notification by email. The notifications are to be recommended for occasional users of an application or for users working on multiple workflow applications.
- Determine the states for which it is necessary to select the actor or the actors having to carry out the following operation. This selection can be associated with an email notification. The notifications are recommended either, for user's occasional application, or for users working on several workflow applications.
- Indicate times of operations, if necessary,
- Specify the structure (sections and fields) documents.

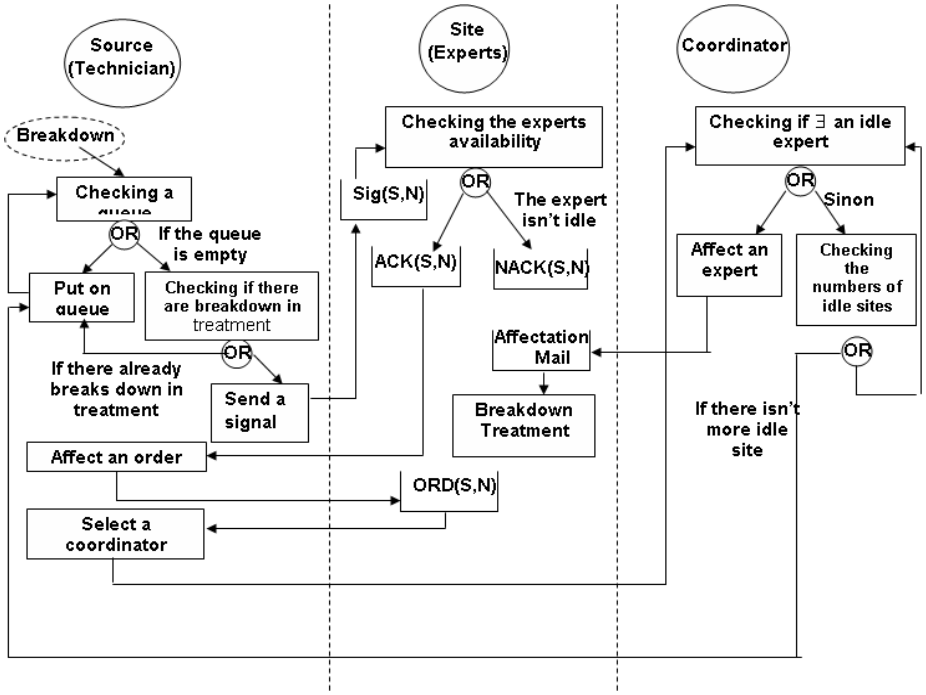
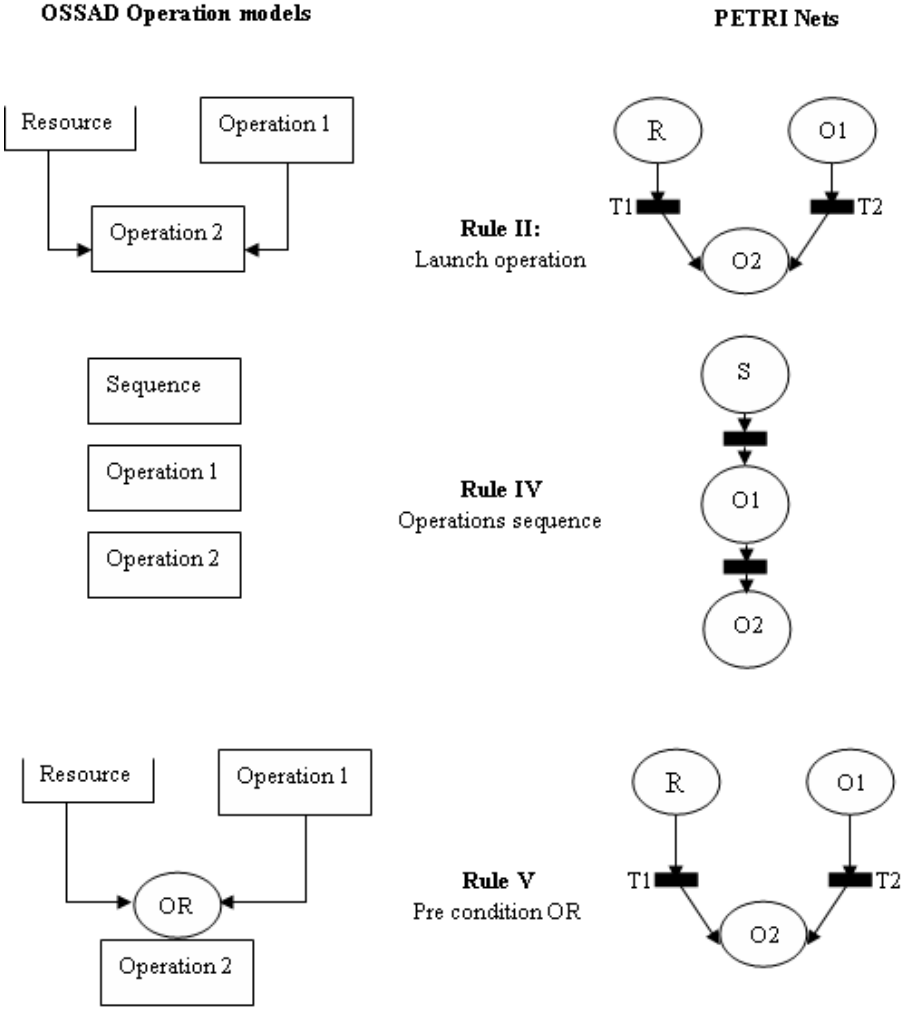


Fig 5. OSSAD operation Model of the group creative process - an expert attribution - additional breakdown treatment

4.2. Generation of the Petri networks from OSSAD models

It was often reproached to workflow models, the absence of possibility of checking and of simulation due mainly to the lack of formalism and the perspective model (workflow) of OSSAD is not safe from these analysis, since, from the motivation of its designers, OSSAD is a relatively simple method, and whose interpretation is little formalized [13]. To overcome these deficiencies, Van Der Aalest [14] introduced the concept of WF NET, the workflows based on a modeling by Petri nets.



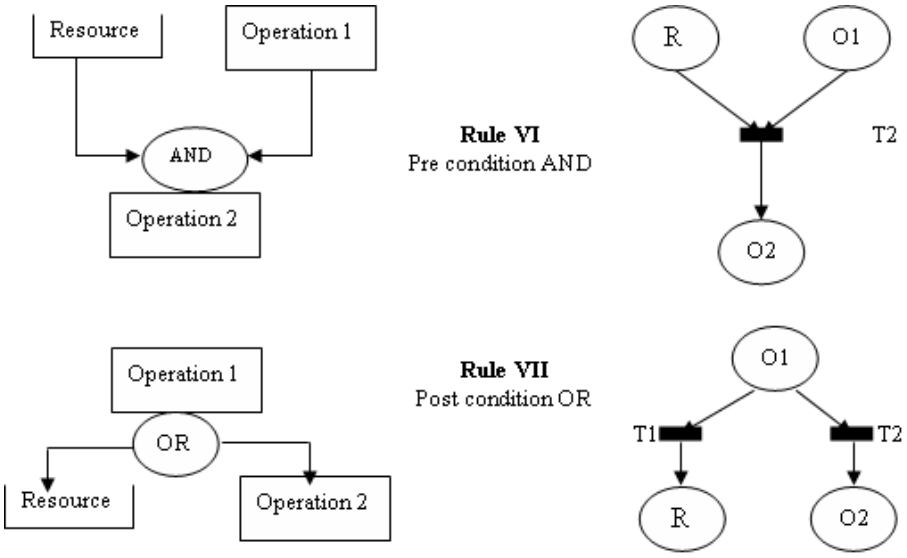


Fig 6. Some transformation rules from OSSAD Models to Petri net [17]

The argumentation of Van Der Aalest was based on the fact that the Petri net are an intuitive graphic language and who led to workflow models whose definition is clear and precise[15]. Moreover, these last years, much of research were carried out on the mathematical properties of the various Petri Net varieties, which has to generate an expansion of methods and techniques for Petri Net analysis which were of a great contribution to modeling workflow. Since these techniques, allows proving the model properties (promptness, conflict, invariant…) and analyzing its performances through various analysis and simulation tools.

However, we cannot allow leave to the end-user the load of Petri nets creating, that are rather the prerogative of expert in the field of data-processing and mathematical modeling. This is why; we will use the theoretical bases of the "OSSAD" formalism and the ten rules defined by Chappelet and Snella [17] for the transformation from OSSAD operation model to Petri Net, in order to generate automatically Petri networks. The **states** (of a Role, a Resource or a Tool) will be interpreted like **"places"** and the **Operations** (of this same model) fig (8), usually represented by squares will be interpreted like **"transitions".** The obtained Petri nets will have a well defined syntax and a logical interpretation. They will make it possible to represent them inter dependences between operations in terms of sequence, availability, parallelism or simultaneity (AND), of conflict or exclusiveness (OR), fig (6).

Note: In certain cases, to simplify, the places (circles) are illustrated only in beginning and end, like for "OR".

4.3. Verification of the Petri Net properties

The evolution of Petri Net is done by crossing of transitions. When during its evolution, certain transitions are never crossed, that indicates that modeled system will not run. There is thus a problem on the systems design level. The idea is to be systematically able to detect this phenomenon, by the Petri Net model properties analysis, in order to have a tool of assistance of the systems design. To check the different properties from our models, and to simulate the circulation of the token in order to detect possible structural conflicts, we chose used PetriParc[1] application, fig (7).

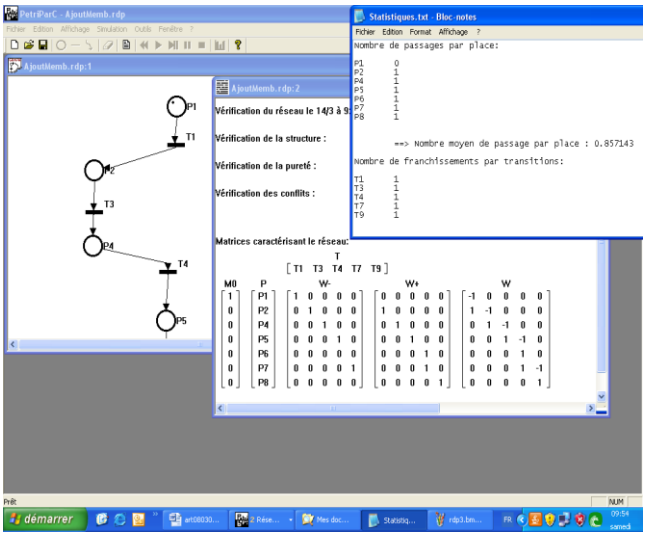


Fig 7. Verification of Petri Net with PetriParc.

[1] www.univ-valenciennes.fr/GDR-MACS/outils.php?id=15

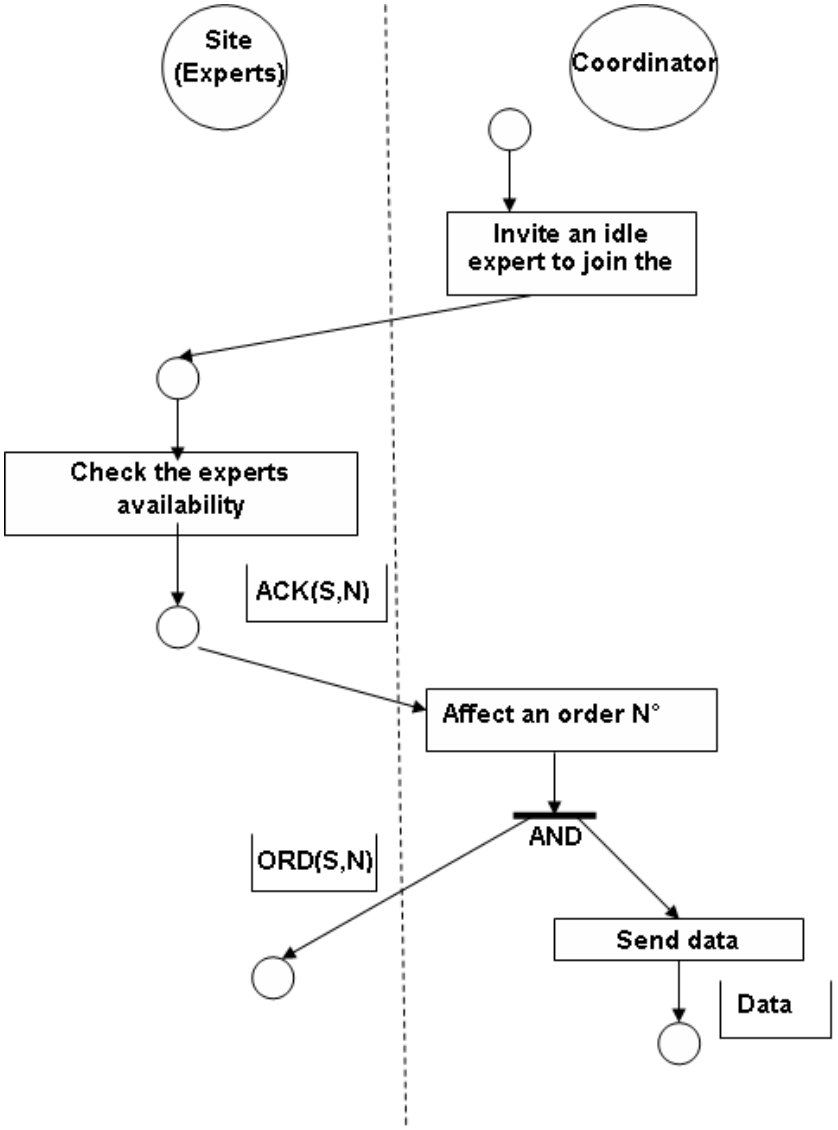


Fig 8. OSSAD Petri net corresponding to the operation model of "joining of a new member by invitation"

5. Conclusion

The Maintenance in general and diagnoses in particular, are processes requiring a great **coordination**, an intense **collaboration** since the actors are

divided as well geographically as temporally. It seems to us legitimate then to choose to implement a workflow system to assist the cooperative work of a maintenance team. It is however necessary to take into account that the workflow, contrary to the other traditional computer applications, does not contribute to the computers work automation, but the accomplished human work automation through multiple cooperation and coordination interactions. Beyond the computers treatments, the workflow attempts to assist the man in his interactions with other men via the computers. Data processing for communication (which includes the workflow), is interested in the human interactions and the subjacent behaviors of communication.

The processes profiting the most from these technologies are thus those based on the communication and collaboration for the achievement of the process objective, in our case, maintenance; this nuance led us to choose a double language of modeling OSSAD/Petri Net for the specification of Workflow, thus enabling us to have at the same time a facility of use thanks to the first one, and a precision of formulation as well as opportunity of analysis and simulation from the second one.

References

1. E. Levrat, B. Salzemann, F. Clanché TELMA : Plate-forme d'intégration de télémaintenance pour l'enseignement et la recherche ». CETSIS, Nancy, (2005).
2. T. Bangemann, X. Rebeuf, D. Reboul, A. Schulze, J. Szymanski, J-P. Thomesse, M. Thron, N. Zerhouni. : "PROTEUS : Creating distributed maintenance systems through an integration platform". Computers in Industry 57, 539–551, (2006)
3. D. Saint-Voirin, C. Lang, D. Saint-Voirin, Christophe Lang, Herv´e Guyennet, N. Zerhouni: "Scoop Methodology : Modeling, Simulation And Analysis For Cooperative Systems". JOURNAL OF SOFTWARE, VOL. 2, NO. 4, (2007).
4. D. Saint-Voirin: « Contribution à la modélisation et à l'analyse des systèmes coopératifs : application à la e-maintenance ». Thèse de Doctorat, LIFC, Université de Franche-Comté, (2006).
5. Ellis, Clarence et J. Wainer : A Conceptual Model of Groupware. Actes de la conférence ACM Computer Supported Cooperative Work (CSCW'94), , pages 79-88, ACM Press. (1994)
6. C. Ellis E, S. Gibbs, G. Rein, : Groupware: Some Issues and Experiences (Journal, Communications of the ACM (CACM), volume 34, number 1, pages 38-58, ACM Press. (1991)
7. J. Grudin : CSCW: History and Focus. Jounal, IEEE Computer, volume 27, number 5, pages 19-26, IEEE. (1994)
8. M., Landauer T.K., Prabhu P : Handbook of Human-Computer Interaction. Ed. by Helander Elsevier Science Publishers, Netherlands (1997).
9. C. Kolski : Interfaces homme-Machine, application aux systèmes industriels complexes (2 ème édition revue et étendue). Paris : Editions Hermes, (1997).
10. S. Nurcan « Analyse et conception de systèmes d'information coopératifs » - Université Paris 1 –Sorbonne. (1996).
11. J-L. Chappelet, A. Le Grand « Modélisation, simulation et génération d'applications de workflow pour l'Internet » / Working paper de l'IDHEAP (2000).

12. M. Boussejdra « la gestion da l'information pour la télémaintenance et le télédiagnostique coopératif ». Mémoire de DEA. LIFC (2001).
13. I. Attali , R. Bastide, M. Blay, A. Dery, P. Palanque « Spécification formelle et approche objet pour les applications Workflow » (1997).
14. Van der Aalst : "Three Good reasons for Using a Petri-net-based Workflow Management System". In S. Navathe and T. Wakayama, editors,Proceedings of the International Working Conference of Information and Process Integration in Enterprises (IPIC'96), pages 179–201, Cambridge, Massachusetts, (1996).
15. Van der Aalst : "Verification of Workflow Nets". In P. Azema and G. Balbo, editors, Application and Theory of Petri Nets 1997, volume1248 of Lecture Notes in Computer Science, pages 407–426. Springer-Verlag, Berlin, (1997).
16. Muller et Al: "On the concept of e-maintenance: Review and current research". (Reliability engineering and system safety journal) Elsevier 2007
17. J.L. Chappelet, J.J. Snella : « Un langage pour l'organisation, l'approche OSSAD » Third Edition PPUR 2004.
18. R. Hogan, T. Cesarone, , D. Dragun : "Battle group automated maintenance environement". Proceedings of the 13th annual international ship control systems symposium explores automation in ship control, Philadelphia, USA, 2003.
19. J Campos, O Prakash : " Information and communication technologies in condition monitoring and maintenance—a review". IFAC symposium INCOM06. 17–19 May, Saint Etienne, France, 2005.
20. A Jardine, D Lin, D Banjevic : " A review on machinery diagnostics and prognostics implementing condition-based maintenance". Mech Syst Signal Process 2006.

Karim BOUAMRANE was born in Oran (Algeria) in 1968. He graduated in 1991 from Department of Computer Science, Faculty of Sciences, University of Oran, Algeria. He received his PhD degree in computer science in 2006. He is chief of the Research Group in « Diagnosis, Decision Support System and Human interaction" in LIO laboratory. His research interests include Decision support system in transportation system, production system and risk management, also, data warehouse and elearning.

M.A LAREDJ was born in Mostaganem (Algeria) in 1980. He graduated in 2002 from Department of Computer Science, Faculty of Sciences, University of Oran, Algeria. He received his Post graduation degree in computer science in 2005. He prepare his Phd thesis within « Diagnosis, Decision Support System and Human interaction" Research Group in LIO laboratory. His research interests include collaborative work, workflow, BPR and e-maintenance.

Received: March 26, 2010; Accepted: January 06, 2011.

DOI: 10.2298/CSIS100827002S

Access Control Framework for XML Document Collections

Goran Sladić, Branko Milosavljević, Zora Konjović, and Milan Vidaković

Faculty of Technical Sciences, University of Novi Sad, Trg D. Obradovića 6
21000 Novi Sad, Serbia
{sladicg, mbranko, ftn_zora, minja}@uns.ac.rs

Abstract. It is often the case that XML documents contain information of different sensitivity degrees that must be selectively shared by user communities. This paper presents the XXACF (eXtensible Role-Based XML Access Control Framework) framework for controlling access to XML documents in different environments. The proposed access control model of XXACF is described. The framework represents an improvement over the existing systems and enables defining context-sensitive access control policies on different priority and granularity levels, the enforcement of access control for different operations on XML documents, as well as different ways of access control enforcement for the same operation.

Keywords: access control, RBAC, context, XML.

1. Introduction

Access control is only one aspect of a comprehensive computer security solution, but also the one of its most important segments. It provides confidentiality and integrity of information. In the role-based access control (RBAC) model, access to resources of a system is based on a role of a user in the system [21]. The basic RBAC model comprises the following entities: *users*, *roles* and *permissions*, where permissions are composed of *operations* applied to *objects*. In RBAC, permissions are associated with roles, and users are made members of roles [21].

The growth of use of XML as a format for data modelling and interchange accentuates the issue of access control to XML documents. An XML document may contain data with different levels of accessibility. eXtensible XML Role-Based Access Control Framework (XXACF) [39] provides the means of defining access control policies and access control enforcement based on the RBAC model. Access control policies in XXACF may be defined on different priority and granularity levels and they may be content dependent, thus facilitating efficient management of access control. The access control policy in XXACF may be separately defined for each operation on an XML document. The concept of context-sensitive access control enables customization of access control policies depending on the environment where XXACF is being used. Therefore, XXACF can be deployed in various environments.

The rest of the paper is structured as follows. Section 2 reviews the related work. Section 3 presents the XXACF data model. The extended DOM model is presented in Section 4. Section 5 describes the procedure of access control enforcement. Section 6 concludes the paper and outlines further research directions.

2. Related Work

In [29] the authors present a provisional authorization model that provides XML with element-wise access control mechanism. They have formalized a provisional authorization model that adds extended semantics to traditional authorization models. The proposed model integrates several security features such as authorization and non-repudiation in unified XML documents and enables the authorization initiator not only to securely browse XML documents but also to securely update each document element.

In the paper [11] access control for XML documents in a workflow environment is presented. The access control policies of the workflow system are based on the RBAC model. There are two types of access control policies [11]: policies which grant access (operation) and policies which deny access (operation) on the object. The operations that are supported are as follows: *read*, *edit*, *add* and *delete* [11]. Each process in the workflow system implies manipulation of a certain number of XML documents. The definition of the given process defines which roles can execute that process, i.e. its sub-processes. It is possible to create access control policies not only on the document level but also on the document fragment level [11].

In the paper [25] the author defines a security model for a native XML database which supports XUpdate language. Since the model is implemented in a native XML database supporting XUpdate, a privilege that each XUpdate operation requires for completion needs to be specified. Presented mode is inspired by the SQL security model.

X-RBAC [9] [10] is a system for controlling access to XML documents in web service-based systems. The access control in this system is based on the RBAC model. Access control policies depend on the user's session context and the content of the documents being accessed [9]. Depending on the document content, the access control policies can be specified on four granularity levels: the conceptual level, the document schema level, the document instance level and the document element level [9]. However, policies that deny access cannot be specified. The supported operations are *reading*, *writing*, *navigating* (reading the referenced data) and *all* (supports all three previous operations). The propagation level down through hierarchy can be defined for each access control policy. The three levels of propagation are supported: (a) without propagation, (b) first level propagation and (c) cascade propagation [10].

XML ACP (Access Control Processor) [16] [17] represents a system for access control for XML documents in web-based applications. The system works as a plug-in to the existing web server technology [16]. The access control is

possible on several levels of granularity: the DTD level, the document instance level, the specific document element/attribute level. The access control policies can be classified into eight priority groups. Propagation of the access control policies down through the hierarchy is supported. The policies can grant or deny access [16] [17]. In access control policy definitions, user identification is based on the user ID or the user group ID and/or the location (computer or computer network) ID being the origin of the request to access a document [16] [17].

The Author-X system [6] [8] provides for defining access control policies on different granularity levels, which can grant or deny access. The controlled propagation of access control policies is provided, where policies defined for a document or DTD can be applied on other semantically related documents or DTDs on different granularity levels [8]. There are two working modes: *pull* and *push* [6]. In the pull mode, the user explicitly requests access to a document. After reception of the request, Author-X forms a document that will contain only the data visible for the user, i.e. the data that the user can change. After the change has been done, the user sends the changed content to the system and the system verifies if the changed content is in accordance with security rights [6]. In the push mode, the system periodically sends documents to all users. Although the same document is sent to all users, the specified security rights are enforced by encrypting parts of the document with different keys for different policies. Each user possesses keys which are available to her or him according to the specified security rights [6].

Qi et al. [35] propose an approach to XML access control through rule functions that are managed separately from documents. The key idea is to encode the access control rules as a set of rule functions that separately perform the actual access evaluation. A rule function is an executable code fragment that encapsulates the access rules and is shared by all documents of the same type. According to authors, the novelties of this model are the high scalability and high performance.

Fundulaki and Maneth [24] propose language for specifying access control on XML data in the presence of update operations. The update operations used in this model are based on the W3C XQuery Update Facility specification. Alternative language that supports access control annotations at the level of the XML DTD is also presented.

In [36] access to a document and its parts can be defined based both on the current document content and on the history information that captures the operations performed on that document. Moreover, the history information includes the source of the parts of a document that were transferred from different documents.

Byun and Park [12] propose two phase filtering scheme for access control enforcement mechanism. The first phase filtering is to abstract only necessary access control rules based on a user query. The second phase filtering is to modify an unsafe query into a safe one. Query modification is the development

of an efficient query rewriting mechanism that transforms an unsafe query into a safe yet correct one that keeps the user access control policies.

An and Park [3] present access control labeling scheme for efficient secure query processing under dynamic XML data streams.

Knowledge based formal approach to ensure the security of web-based XML documents is presented in [4]. Given approach is based on a high level language to specify an XML document and its protection authorizations.

The focus of the paper [15] is how to control access to XML documents, once they have been received. The paper describes how certain access control policies for restricting access to XML documents can be enforced by encrypting specified regions of the document. The regions are specified using XPath filters and the policies based on the hierarchical structure of XML document. Objects are ordered using containment of their respective XPath expressions. They are encrypted with a number of different keys based on the relative seniority of objects. A user is supplied with a master key enabling her/him to decrypt those objects for which she/he is authorized.

Miklau and Suciu [31] propose framework for enforcing access control policies on published XML documents using cryptography. In this framework the owner publishes a single data instance, which is partially encrypted and which enforces all access control policies. The owner enforces an access control policy by granting keys to users. A client can access the data conditionally, depending on the keys she/he possesses. The client does not need to decrypt the entire data instance; it can access to data selectively using a query language. Authors also present declarative language for access control policies and extension to XQuery to support selective access to a document.

XACML (eXtensible Access Control Markup Language) [32] provides a general purpose access control language. It describes both an access control policy language and a request/response language. The policy language is used to express access control policies (who can do what when). The request/response language expresses queries about whether a particular access should be allowed (requests) and describes answers to those queries (responses). XACML also suggests a policy authorization model to guide implementers of the authorization mechanism. The hierarchical resource profile of XACML [33] specifies how XACML can provide access control for resources (including XML documents) that are organized as a hierarchy. The core and hierarchical role-based access control (RBAC) profile of XACML [34] defines a profile for the use of XACML to meet the requirements of RBAC. Haidar et al. [1] analyze RBAC profile of XACML and they identify several limitations of proposed profile.

Various definitions of context have been proposed in the literature [2] [13] [19] [23] [37]. Broadly, the notion of a context relates to the characterization of environment conditions that are relevant for performing appropriate actions in the computing domain. In order to realize the fine grained access control many context-sensitive access control models have been proposed. Many authors [20] [27] [38] [41] propose a context-based access control model for web services. Their approaches grant or adapt permissions to users based on a set

of contextual information collected from environments of the system. Covington et al. [14] introduce the notion of environmental role, and provide a uniform access control framework that can be used to secure context-sensitive applications. In the paper [22] authors showed the use of context information and its quality indicators to grant access permissions to resources. Georgiadis et al. [26] discuss the integration of contextual information with team and role-based access control. The influence of temporal constraints to access control is probably the most thoroughly analyzed in [5] [30], while the influence of geospatial constraints is presented in [7] [18].

By analyzing the previously mentioned XML access control frameworks, we notice that a significant number of them do not support the RBAC-based access control, but are based on different access control models. Although the literature recognizes a significant number of context-dependent access control models, the impact of context to XML access control is only partial. All presented XML access control frameworks support the access control enforcement for reading documents, but not for creating, updating and deleting documents. To the best of our knowledge, research on access control to XML document in the presence of XML Schema is very poor. The presented models usually implement the reading operation by pruning parts of documents for which users are not authorized. This approach may cause that the newly formed document is not in accordance with its schema. Cases when a user requests that the documents that she/he receives are in accordance with theirs XML Schema are only partially considered.

XXACF, the system presented in this paper, has the following notable improvements over the aforementioned XML access control systems:

- Access control is based on the RBAC model with the support for the role hierarchy
- Context-sensitive access control enforcement is supported
- Definition of granting and denying access control policies on different priority and granularity levels and document content-dependent policies are supported
- Support for separate access control enforcement for different operations on documents
- Different ways of implementing read operation on documents in order to provide: (a) pruned reading, (b) reading when t is required that a document is in accordance with its schema after access control enforcement and (c) specific reading to improve performance in case of a large number of accesses to a document

3. The XXACF Access Control Policies Model

The XML Schema in Figure 1 defines the structure of access control policies, which is based on policies described in [29] supporting policy conditions. Access control policies described in [29] are extended in order to support role hierarchies, to provide priorities of access control policies as defined in [16] [17]

and to support arbitrary operations for which policies are defined. We use our schema for brevity, although this model can be represented through the XACML language.

Each access control policy is defined by the `<policy>` element to which the unique identifier (the `id` attribute) is assigned. The policy (`<permission>`) permits or denies the subject to execute the operation (`<operation>`) on the object (`<object>`). Since access control policies are based on the RBAC model, the subject for which the policy is defined represents the role. The object of the access control policy is a document schema or a document instance identified by its unique ID. Policies defined for a document schema are applied to all document instances of that type (all documents that are valid according to the schema), while policies defined for a particular document instance are applied only to that instance. If the policy is defined only for the particular document/schema fragment, it is necessary to specify the XPath expression which selects that fragment. Also, by using XPath expressions with conditions, an object (policy) which is content dependent, may be specified.

XXACF can support context-sensitive access control and dynamic capturing of context information through `<condition>` element. If the condition is satisfied, the access control policy will be applied. Otherwise, it will not be the case. Each condition consists of the name of a logical operation (*not*, *and*, *nand*, *or*, *nor*, and *xor*) (attribute `operation`) and subconditions and/or predicates (`<predicate>` element). A predicate is a function which returns a Boolean value. The value calculated by applying a specified logical operation on return values of subconditions and predicates represents the return value of the condition. Functionality of a predicate should be properly implemented. This facilitates the implementation of the specific access control which is not supported through the RBAC model. For example, insertion of a digital signature into a document is allowed only if the signature is valid and is generated using the valid key of the logged user. This verification can be realized by implementing a predicate that will verify a signature of the document using the certificate of the logged user and also verify if the certificate is revoked. By using a predicate where the return value depends on the state of the environment (e.g. access to the document depends on the current state of the process which the document belongs to), it is possible to implement context-sensitive access control.

Although the standard RBAC model supports only granting policies, results in the literature (see Section 2) identify a need for denying policies to achieve more efficient security administration. The permission is described by the element `<permission>` shown in Figure 1. The permission can be dual, granting access (the value of the `type` attribute is "*grant*") or denying it (the value of the `type` attribute is "*deny*"). In order to avoid specifying explicitly the policy for each entity, propagation of policies is enabled, starting from the entity specified by the `<object>` element down or up the hierarchy. The `propagation_direction` attribute defines the propagation direction. If the attribute value is "*down*", the propagation is directed from the specified object downwards along the hierarchy. On the other hand, if the value is "*up*", it is directed from the specified

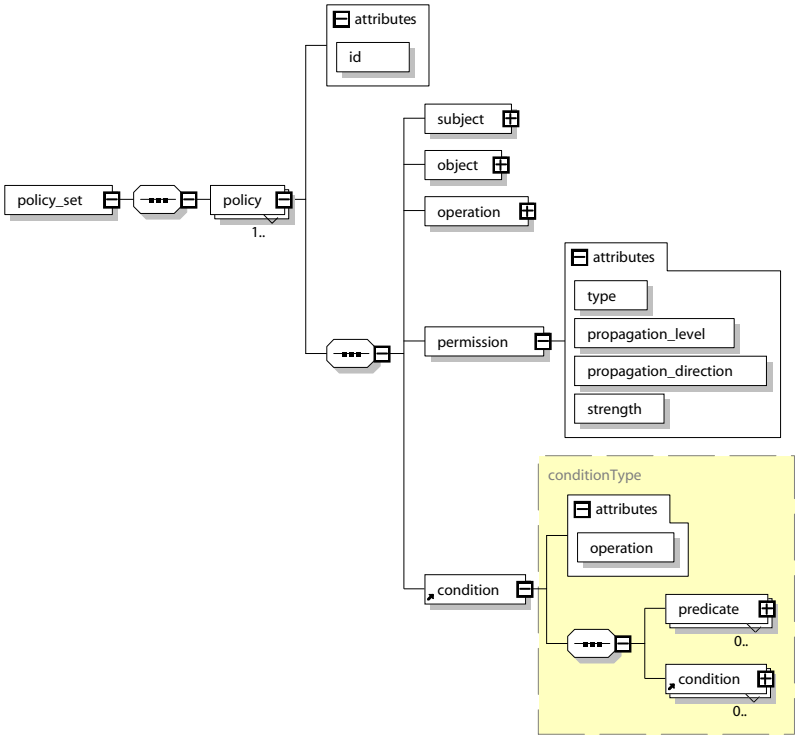


Fig. 1. XML Schema of access control policies

object upwards. The level of propagation, i.e. the maximum number of hierarchy levels where the propagation is performed, is specified by the attribute `propagation_level`. The propagation level can be arbitrary, it can include a certain number of levels, or with no propagation at all. Access control policy strength is defined by the `strength` attribute as defined in [17]. Values of the `strength` attribute can be "*normal*", "*hard*" and "*soft*". The value "*normal*" can be specified if the object of the access control policy is a document schema or a document instance, "*hard*" is specified only for a document schema and "*soft*" if the object is a document instance. Detailed explanation of the semantics of the `strength` attribute and it's values can be found in [17].

Access control policies defined for a descendant role have a higher priority than the policies defined for an ascendant role. The priority of access control policies defined for the same role is determined according to the object they are related to (a document schema or a document instance), to the strength of the policy and also to the propagation level of the policy. On the basis of these elements it is possible to define eight priority levels presented in Table 1. We believe that this approach for determining priorities is more applicable in practice than the explicit priority assignment, because the explicit assignment does not necessary have the semantics of the priority values.

Table 1. Priority levels of access control policies

Priority level	Schema/Instance	Propagation	Strength
1	schema	no	hard
2	schema	yes	hard
3	instance	no	normal
4	instance	yes	normal
5	schema	no	normal
6	schema	yes	normal
7	instance	no	soft
8	instance	yes	soft

4. The Extended DOM Model

The process of access control enforcement for XML documents in XXACF is performed on DOM (Document Object Model) of the document. In order to achieve access control, the DOM model is extended with necessary functionality.

The diagram in Figure 2 shows the classes which represent a node in a document tree. Each implementation of the `XACNode` interface contains the `MarkMap` object. It contains all access control policies applied on that node (Figure 2 shows the example for `XACElementImpl`). The role for which the access control policy is defined is used as a key to the hash map contained in the `MarkMap` class. Hash map values are instances of the `MarkMatrix` class. This class contains access control policies defined for the given role and applied to that node.

`MarkMatrix` distributes access control policies on one of the `Set` sets of the `matrix` matrix. The row index depends on if the access control policy grants or denies access to that node. The column index represents the priority level of the access control policy. Therefore, the matrix has two rows and eight columns.

Each of these sets contains instances of `MarkItem`. `MarkItem` contains the access control policy and the distance from the root node selected by the given policy to the node to which that `MarkItem` instance belongs.

For each access control policy, the class `Marker` determines which nodes of the tree are selected by the object of the policy and applies the policy to these nodes.

To avoid implementation of the functionality defined by the interfaces of the DOM model (interfaces from the package `org.w3c.dom`), *Apache Xerces* implementation of the DOM model is used. An example of implementation of the `XACElement` interface is shown in Figure 2. The class `XACElementImpl` represents implementation of `XACElement`. `ElementImpl` is Xerces implementation of the `Element` interface. By extending this class, implementation of the functionality defined by `Node` and `Element` interfaces is avoided. Other specializations of the `XACNode` interfaces are implemented in the same way.

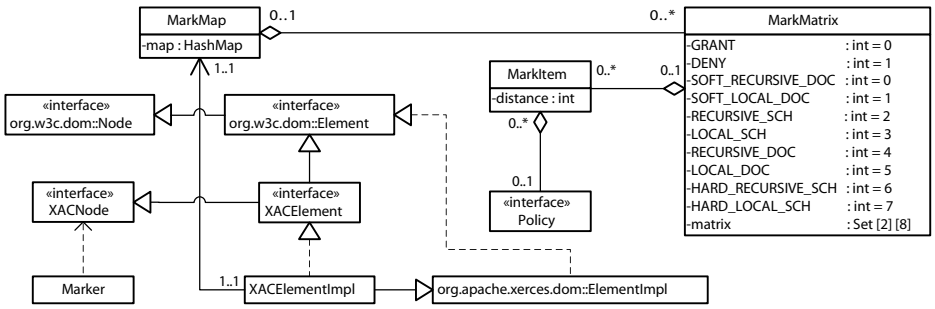


Fig. 2. Classes for marking a node

5. Access Control Enforcement

The process of access control is performed in four steps:

1. selection of the applicable access control policies,
2. marking document nodes,
3. conflict resolution, and
4. execution of the requested operation.

These four steps are described in the following subsections.

The description of the access control enforcement process adopts the following notions.

The set of all document schemas (S) is comprised of all document schemas defined in a system, i.e. $S = \{St_i \mid 0 < i \leq n\}$ where St_i is the i-th document schema and n is the number of schemas in the system. Each St_i consists of the main XML Schema and all XML Schemas that are included in the main.

Let D be the set of all documents in the system and Dt_i be the set of all documents of a particular type t_i (all documents which are in accordance with a certain schema). Then D is defined as follows $D = \bigcup_{i=1}^{n} Dt_i$, where n is the number of different schemas (document types) in the system.

The set of all access control policies is defined as follows $P = PD \cup PS$, where: $PD = \bigcup_{i=1}^{n} PDt_i, PS = \bigcup_{i=1}^{n} PSt_i$, PD is the set of all policies defined for documents, PDt_i is the set of all policies defined for documents belonging to the set Dt_i, PS is the set of all policies defined for document schemas and PSt_i is the set of policies defined for the document schema St_i.

Let R be the set of all user roles, U is the set of all users in the system and O is the set of operations defined in the system.

$markItem$ is defined as a tuple of the form: $markItem = (policy, distance)$, where: $policy \in PDt_i \cup PSt_i$ and $distance \in N_0$ is the distance between the root node selected by the object of the policy and the node to which the access control policy is applied.

5.1. The Selection of the Applicable Access Control Policies

The goal of this task is to find access control policies that will be used for access control enforcement on the document being accessed.

When a document $doc \in Dt_i$ is being accessed in order to execute a certain operation, only access control policies that apply to the document doc from PDt_i and its schema (PSt_i) are loaded from repositories. The access control policies being loaded must be defined for the specified operation and for the roles assigned to the user who performs the operation on that document. The system tests the condition of usage (if the condition is specified) for each loaded access control policy. If the condition is satisfied, the policy will be applied (it is added to applicable policies set, APS); otherwise it will not be applied.

5.2. Marking Document Nodes

Marking of document nodes is a process of determining and applying each policy from the applicable policy set to the nodes selected by that policy.

Depending on if the propagation direction for a policy is down ("*DOWN*") or up ("*UP*") the hierarchy (see Section 3), the appropriate propagation is performed until the maximum level of propagation is reached. A `MarkItem` instance is created for each selected node and added to the appropriate instance of `MarkMatrix` of the node's `MarkMap`.

Upon completing this task, each node will be associated with access control policies to be applied to it.

5.3. Conflict Resolution

Since XXACF supports granting and denying policies it is possible that on some nodes both policy types are applied. In these cases a conflict need to be resolved, i.e. determine whether granting or denying policies will be applied. Our procedure for conflict resolving is presented in Algorithm 1. The $NodeConflictResolve$ function is invoked for each node of the specified subtree to resolve the conflict for the given node.

The first activity of selecting the final access control policies for the given node is conflict resolution according to the "*more specific subject takes precedence*" (MSSTP) principle (invoking the $ResolveConflictByMSSTPPrinciple$ function). According to this principle, access control policies defined for more specific roles have a greater priority than the ones defined for less specific roles (if the role r_i extends the role r_j then r_i has greater priority). For each user's role it is checked if there are policies defined for that role. If the policies defined for that role exist, they are selected. On the other hand, if they do not exist, the role hierarchy is recursively traversed in search of policies.

Next, the conflict resolution using the "*more specific object takes precedence*" (MSOTP) principle is performed on the policies previously retreived. In this step, the policies where the object is nearest to the node which the policy is applied to are selected. First, in the grant policies set (Grant Mark Item Set,

Algorithm 1 Conflict resolution for a node

```
NAME: NodeConflictResolve
INPUT: node - node for which conflict resolving is preformed
URS ⊆ R - user's roles
defaultConflictResolution - default conflict resolution for document
defaultPolicy - default policy
OUTPUT: node - node with determined final policies

Let FPS be the set of final policies for node
Let markMap be the mark map of node
markMatrix := CreateEmptyMarkMatrix()
markMatrix := ResolveConflictByMSSTPPrinciple(markMap, URS, markMatrix)
collIndex := FindTheHigestNonEmptyColumn(markMatrix)

{If there is such a column, resolve conflict for policies in that column using
the MSOTP principle}
if collIndex ≠ −1 then
  Let GMIS be the mark item set with grant policies in markMatrix at the
  column collIndex
  GMIS := KeepPoliciesWithMSO(GMIS)
  Let DMIS be the mark item set with deny policies in markMatrix at the
  column collIndex
  DMIS := KeepPoliciesWithMSO(DMIS)
  MIS := ResolveConflictByMSOTPPrinciple(GMIS, DMIS)

  if HasGrantPolicies(MIS) ∧ HasDenyPolicies(MIS) then
    if defaultConflictResolution = grant_takes_precedence then
      FPS := GetGrantPolicies(MIS);
    else
      FPS := GetDenyPolicies(MIS);
    end if
  else if HasGrantPolicies(MIS) then
    FPS := GetGrantPolicies(MIS);
  else if HasDenyPolicies(MIS) then
    FPS := GetDenyPolicies(MIS);
  else
    FPS := {defaultPolicy} {No policies}
  end if
else
  FPS := {defaultPolicy} {No policies}
end if
SetFinalPolices(node, FPS)
```

$GMIS$) only policies with the minimal distance are kept and the remaimning policies are removed. The same action is preformed on the denial policies set (Deny Mark Item Set, $DMIS$). The result of conflict resolution using the MSOTP principle is the one of these two sets ($GMIS$ or $DMIS$) which has the minimal distance, or the union of these two sets ($GMIS \cup DMIS$) if the distance is the same.

At the end of the conflict resolution process it is checked if there are access control policies that grant and deny access. If there are both, the default conflict resolution ($defaultConflictResolution$) is verified. In the case that in the selected priority level all access control policies either grant or deny access, the final access control policies either grant access, or deny it. If no policies are applied to a node, then depending on the value of the *default policy*, access to that node will be granted or denied.

After the conflict resolutiont each node is associated with its final access control policies, either granting or denying access. There may be more than one final policy associated with a node.

5.4. Executing Operations in XXACF

XXACF supports the following operations on XML documents:

- updating documents (**adding** new nodes, **deleting** and **changing** (replacing) the existing ones) and
- reading documents. The reading operation is implemented in three different ways: **pruned reading**, **fake reading** and **encrypted reading**.

Modifying Operations The complete process of adding new nodes into a tree along with access control enforcement is defined by Algorithm 2. Each new subtree to be *inserted* into the document tree is temporarily inserted. Then, the selected access control policies are applied and the conflict resolution process on the *inserted nodes* is performed. If the final access control policies on each node of the inserted subtree grant access, the insertion is allowed. If insertion of any node of the inserted subtree is not allowed, the subtree is removed from the document tree.

Algorithm 2 Add operation

NAME: AddOperation
INPUT: $session$ - user's session
$doc \in Dt_i$ - document being accessed
$ANS = \{(node, pos) \mid node$ - root node of subtree, pos - position in document where node should be inserted$\}$ - set of subtree root nodes which are added to the document
$URS \subseteq R$ - user's roles
OUTPUT: doc - document with new nodes if insertion of new nodes is allowed

{Conditialnally insert all new nodes in the document on the specified position}
for each $item \in ANS$ **do**
  doc_i := InsertNodeInDocument(doc_i, $item.node$, $item.pos$)
end for

sch_i := GetDocumentSchema(doc_i)
APS := GetApplicablePolicies($session$, doc_i, sch_i, add, USR)
$xdom$:= MarkDocument(doc_i, APS)

for each $item \in ANS$ **do**
  $xdom$:= ResolveConflict($xdom$, $item.node$, URS)
  if ¬InsertionOfSubtreeAllowed($xdoc$, $item.node$) **then**
    $xdoc$:= RemoveSubtree($xdoc$, $item.node$)
  end if
end for
doc := Transform($xdom$)

In the same way as in the case of the insertion operation, the **delete** operation of the subtree is allowed if deletion of all nodes of the subtree is allowed,

i.e. if the final access control policies of each node of the subtree permit this operation.

For **replacement** of a subtree by another subtree, it is necessary that access control policies allow replacement of the nodes, i.e. that it is allowed to replace one of the nodes and that replacement with the given node is allowed. Hence, two sets of access control policies have to be defined for the operation of replacing a subtree. The first set defines the precondition (if the node can be replaced). The second set defines the postcondition (if replacement with the given node is allowed).

Reading Operations XXACF currently supports three types of reading operations: pruned reading, fake reading and encrypted reading.

Pruned reading provides only reading of those parts of the XML document that are allowed to be read by the user - the content for which the user has no read authorization is removed. Given the extended DOM model of a document, this process is performed by removing nodes of the document tree for which access is not granted. If a node, for which access is not granted, is a leaf, it is removed from the tree. If a node with denied access is not a leaf (hence it is an XML element) then, in order to preserve the document structure, it is not removed from the tree, but its attributes which the user is not granted to read are removed.

The **fake reading** operation processes the document in such a way that the parts of the document not granted to be read are replaced by fake (dummy) values. The purpose of this type of reading operation is to obtain the document which is in accordance with its XML schema. Since the XML Schema standard supports the large number of data types and defining new ones, generating fake values according to the given data type may be very complex. We have opted for using the approach that multiple dummy value generators may be implemented and integrated into XXACF, each targeting a specific data type.

The pruned reading and dummy reading operations are executed on the user demand, i.e. on each request for reading a document. In the case of the large number of accesses to documents for reading, using the previously described reading operations may seriously diminish the system performance.

An effective alternative to these approaches is to use **encrypted reading** - by creating a new document based on the original one, according to the access control policies defined for the original document. Users can access only the parts of the new document for which they are authorized. One of the methods to form that kind of document is to use cryptographic techniques based on keys. According to access control policies, different parts of a document are encrypted with different keys. A user possesses only those keys that enable him or her to decrypt the parts of the documents that she or he is allowed to access. The major problem for this type of reading is to determine which document parts will be encrypted by which key. The simplest approach is to encrypt each document node with a different key, while this key is accessible only to users authorized for access to that node. This approach is simple for implementation,

but can cause generation of a large number of keys. Our solution to this problem is to determine *role groups*, where each group consists of all roles to which access to some node(s) is granted. One key is generated for each role group; all nodes for which that group has the access right are encrypted by that key.

The XML Encryption specification allows only encryption on the element level and it is possible to encrypt the whole element or its content only [28]. If an attribute role group differs from the role group of its element, it is necessary to encrypt that attribute with another key. In order to enable attribute encryption and maintain conformance with the XML Encryption specification, it is necessary to transform it into the element. The similar case occurs if it is necessary to encrypt an element content (child nodes) with different keys. Since it is not possible to encrypt the whole element with one key, it is necessary to transform all attributes to subelements of the given element. For the same reason, there are situations when all non-element subnodes must be transformed.

Algorithm 3 Creation of an encrypted node set

```
NAME: CreateEncryptionDOM
INPUT: node - currently processed node
ENS - contains root nodes of subtree which are encrypted with the same key in
the previous call
OUTPUT: node - modifyed node, if required in order to enable encryption using
XML Encryption
ENS - set which contains root nodes of subtree which are encrypted with same
key

Let ANS be attributes set of the node
Let CNS be child nodes set of the node
if ANS ≠ ∅ ∨ CNS = ∅ then
  if ANS ≠ ∅ then
    for each attr ∈ ANS do
      CreateEncryptionDOM(attr, ENS) {Recursive call}
    end for
  end if

  if CNS ≠ ∅ then
    for each childNode ∈ CNS do
      CreateEncryptionDOM(childNode, ENS) {Recursive call}
    end for
  end if

  if SameRoleGroup(node, ANS, CNS) then
    RemoveNodes(ANS, ENS)
    RemoveNodes(CNS, ENS)
    AddNode(node, ENS)
  else
    node := TransformNode(node, ENS)
  end if
else
  AddNode(node, ENS)
end if
```

Determining role groups is defined by Algorithm 3. It describes the our procedure for both determining the set of the root nodes of the subtrees in which each node should be encrypted by the same key, and possible subtrees transformation.

Since, the secret keys are stored into the document it is necessary to encrypt those keys with the public keys of users who will read the document using the asymmetric encryption procedure. Only secret keys assigned to user's roles are encrypted with the public key of each user (i.e., keys that belong to role groups which contain one or more user's roles). Then a user will be able to decrypt only secret keys which are encrypted with her/his public key. In this manner, a user can access only secret keys assigned to her/his roles and therefore she/he can decrypt only document fragments which she/he is authorized for.

6. Conclusion

This paper presents the main features of eXtensible XML Role-Based Access Control Framework (XXACF).The proposed access control model provides access control representation according to the RBAC model and enables definition of the context-sensitive access control. It allows specification of the access control policies on the document schema, instance, and fragment levels. Also, content-dependent access control policies specification is possible. XXACF provides access control enforcement for different operations on a document, as well as the possibility of different ways of access control enforcement for the same operation. The presented model separates XML documents from RBAC components and provides independent design and administration of access control policies.

The most notable improvements over the systems reviewed in Section 2 include: (a) context-sensitive access control based on the hierarchical RBAC model, (b) document-dependent definition of access control policies on different priority and granularity levels, and (c) support for separate access control enforcement for different operations on documents and different ways of implementing the same operation.

The XXACF prototype implementation is verified on a document-centric workflow system based on XML documents. The presented prototype implementation represents the proof of the proposed model practical value. Response time for the most documents was satisfactory. The time for access control enforcement is significantly less than the overall time to a document access. Only for a few quite large documents, the access control enforcement time is a significant part of the total access time. XXACF is yet to be verified in different environments. So far we have not considered performance implications outside of the prototype workflow system.

Future work in XXACF development includes the integration with other access control systems, enabling an application that uses XXACF to operate not only with XML documents, but also with data in other formats (such as relational databases). Moreover, we plan to adjust XXACF for access control in distributed agent-based systems like [40]. We think that a formal specification of the context has to be done in order to enable efficient usage of XXACF in different

systems. Functionality of defining constraints of static and dynamic separation of duties (SoD) is also under way.

References

1. Abi Haidar, D., Cuppens-Boulahia, N., Cuppens, F., Debar, H.: An extended RBAC profile of XACML. In: SWS '06: Proceedings of the 3rd ACM workshop on Secure web services. pp. 13–22. ACM, New York, NY, USA (2006)
2. Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M., Steggles, P.: Towards a better understanding of context and context-awareness. In: HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing. pp. 304–307. Springer-Verlag, London, UK (1999)
3. An, D.C., Park, S.: Access control labeling scheme for efficient secure XML query processing. In: Proceedings of the 12th international conference on Knowledge-Based Intelligent Information and Engineering Systems, Part II. pp. 346–353. Springer, Berlin (2008)
4. Bai, Y.: Access control for XML document. In: IEA/AIE '08: Proceedings of the 21st international conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. pp. 621–630. Springer-Verlag, Berlin, Heidelberg (2008)
5. Bertino, E., Bonatti, P.A., Ferrari, E.: TRBAC: A temporal role-based access control model. ACM Trans. Inf. Syst. Secur. 4(3), 191–233 (2001)
6. Bertino, E., Castano, S., Ferrari, E.: Securing XML documents with Author-X. IEEE Internet Computing 05(3), 21–31 (2001)
7. Bertino, E., Catania, B., Damiani, M.L., Perlasca, P.: GEO-RBAC: a spatially aware RBAC. In: SACMAT '05: Proceedings of the tenth ACM symposium on Access control models and technologies. pp. 29–37. ACM, New York, NY, USA (2005)
8. Bertino, E., Ferrari, E.: Secure and selective dissemination of XML documents. ACM Trans. Inf. Syst. Secur. 5(3), 290–331 (2002)
9. Bhatti, R., Bertino, E., Ghafoor, A., Joshi, J.B.: XML-based specification for web services document security. Computer 37(4), 41–49 (2004)
10. Bhatti, R., Joshi, J.B., Bertino, E., Ghafoo, A.: Access control in dynamic XML-based web-services with X-RBAC. In: 1st International Conference on Web Services (2003)
11. Botha, R.A., Eloff, J.H.: A framework for access control in workflow environments. Information Management and Computer Security 9(3), 126–133 (2001)
12. Byun, C., Park, S.: Two phase filtering for XML access control. In: Secure Data Management. pp. 115–130. Springer (2006)
13. Chen, G., Kotz, D.: A survey of context-aware mobile computing research. Tech. rep., Hanover, NH, USA (2000)
14. Covington, M.J., Long, W., Srinivasan, S., Dev, A.K., Ahamad, M., Abowd, G.D.: Securing context-aware applications using environment roles. In: Proceedings of the 6th ACM Symposium on Access Control Models and Technologies (SACMAT). pp. 10–20. ACM, New York, NY, USA (2001)
15. Crampton, J.: Applying hierarchical and role-based access control to XML documents. In: SWS '04: Proceedings of the 2004 workshop on Secure web service. pp. 37–46. ACM, New York, NY, USA (2004)
16. Damiani, E., Samarati, P., di Vimercati, S.D.C., Paraboschi, S.: Controlling access to XML documents. IEEE Internet Computing 5(6), 18–28 (2001)

17. Damiani, E., di Vimercati, S.D.C., Paraboschi, S., Samarati, P.: A fine-grained access control system for XML documents. ACM Trans. Inf. Syst. Secur. 5(2), 169–202 (2002)
18. Damiani, M.L., Bertino, E., Catania, B., Perlasca, P.: GEO-RBAC: A spatially aware RBAC. ACM Trans. Inf. Syst. Secur. 10(1), 2 (2007)
19. Dey, A.K.: Understanding and using context. Personal Ubiquitous Comput. 5(1), 4–7 (2001)
20. Feng, X., Jun, X., Hao, H., Li, X.: Context-aware role-based access control model for web services. Grid and Cooperative Computing GCC 2004Workshops, International Workshop on Information Security and Survivability for Grid 3252, 430–436 (2004)
21. Ferraiolo, D.F., Sandhu, R., Gavrila, S., Kuhn, D.R., Chandramouli, R.: Proposed NIST standard for role-based access control. ACM Trans. Inf. Syst. Secur. 4(3), 224–274 (2001)
22. Filho, J.B., Martin, H.: Using context quality indicators for improving context-based access control in pervasive environments. In: EUC '08: Proceedings of the 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing. pp. 285–290. IEEE Computer Society, Washington, DC, USA (2008)
23. de Freitas Bulcao Neto, R., da Graca Campos Pimentel, M.: Toward a domain-independent semantic model for context-aware computing. In: Proceedings of the 3rd Latin American Web Congress (LA-WEB). pp. 61–70. IEEE Computer Society, Washington, DC, USA (2005)
24. Fundulaki, I., Maneth, S.: Formalizing XML access control for update operations. In: SACMAT '07: Proceedings of the 12th ACM symposium on Access control models and technologies. pp. 169–174. ACM, New York, NY, USA (2007)
25. Gabillon, A.: An authorization model for XML databases. In: SWS '04: Proceedings of the 2004 workshop on Secure web service. pp. 16–28. ACM, New York, NY, USA (2004)
26. Georgiadis, C.K., Mavridis, I., Pangalos, G., Thomas, R.K.: Flexible team-based access control using contexts. In: SACMAT '01: Proceedings of the sixth ACM symposium on Access control models and technologies. pp. 21–27. ACM, New York, USA (2001)
27. Haibo, S., Fan, H.: A context-aware role-based access control model for web services. Proceedings of the IEEE International Conference on e-Business Engineering (ICEBE) 0, 220–223 (2005)
28. Imamura, T., Dillaway, B., Simon, E.: XML encryption syntax and processing. W3C Recommendation (2002), http://www.w3.org/TR/xmlenc-core/
29. Kudo, M., Hada, S.: XML document security based on provisional authorization. In: CCS '00: Proceedings of the 7th ACM conference on Computer and communications security. pp. 87–96. ACM Press (2000)
30. Latif, U., Joshi, J.B.D., Bertino, E., Ghafoor, A.: A generalized temporal role-based access control model. IEEE Trans. on Knowl. and Data Eng. 17(1), 4–23 (2005)
31. Miklau, G., Suciu, D.: Controlling access to published data using cryptography. In: VLDB '2003: Proceedings of the 29th international conference on Very large data bases. pp. 898–909. VLDB Endowment (2003)
32. OASIS: Extensible access control markup language (XACML) v 2.0. OASIS Specification (2005), www.oasis-open.org/committees/xacml
33. OASIS: Hierarchical resource profile of XACML. OASIS Specification (2005), www.oasis-open.org/committees/xacml
34. OASIS: XACML profile for role based access control (RBAC). OASIS Specification (2005), www.oasis-open.org/committees/xacml

35. Qi, N., Kudo, M., Myllymaki, J., Pirahesh, H.: A function-based access control model for XML databases. In: CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management. pp. 115–122. ACM, New York, NY, USA (2005)
36. Roder, P., Tafreschi, O., Eckert, C.: History-based access control for XML documents. In: ASIACCS '07: Proceedings of the 2nd ACM symposium on Information, computer and communications security. pp. 386–388. ACM, New York, NY, USA (2007)
37. Schilit, B., Adams, N., Want, R.: Context-aware computing applications. In: Proc of IEEE Workshop on Mobile Computing Systems and Applications. pp. 85–91. IEEE Computer Society, Washington, DC, USA (1994)
38. Shang, C., Yang, Z., Liu, Q., Zhao, C.: A context based dynamic access control model for web service. In: International Conference on Embedded and Ubiquitous Computing, IEEE/IFIP. vol. 2, pp. 339–343. IEEE Computer Society, Los Alamitos, CA, USA (2008)
39. Sladić, G., Milosavljević, B., Konjović, Z.: Extensible access control model for XML document collections. In: ICETE SECRYPT: Proceedings of the 2nd International Conference on Security and Cryptography. pp. 373–380. INSTICC (2007)
40. Vidaković, M., Milosavljević, B., Konjović, Z., Sladić, G.: Extensible java EE-based agent framework and its application on distributed library catalogues. Computer Science and Information Systems 6(2), 28 (2009)
41. Wolf, R., Keinz, T., Schneider, M.: A model for context-dependent access control for web-based services with role-based approach. Proceedings of the 14th IEEE International Workshop on Database and Expert Systems Applications (DEXA) 00, 209–214 (2003)

Goran Sladić is teaching assistant at the Faculty of Technical Sciences, Novi Sad, Serbia. Mr. Sladić received his Bachelor degree (2002) and Master degree (2006) all in Computer Science from the University of Novi Sad, Faculty of Technical Sciences. Since 2002 he is with the Faculty of Technical Science in Novi Sad. His research interests include information security, context-aware computing and document management. He is the corresponding author and can be contacted at: sladicg@uns.ac.rs

Branko Milosavljević is holding the associate professor position at the Faculty of Technical Sciences, Novi Sad, Serbia since 2008. Mr. Milosavljević received his Bachelor degree (1997), Master degree (1999), and PhD degree (2003) all in Computer Science from the University of Novi Sad, Faculty of Technical Sciences. Since 1998 he is with the Faculty of Technical Science in Novi Sad. His research interests include information retrieval, digital libraries, document management and information security.He can be contacted at: mbranko@uns.ac.rs

Zora Konjović is holding the full professor position at the Faculty of Technical Sciences, Novi Sad, Serbia since 2003. Mrs. Konjović received her Bachelor degree in Mathematics from the University of Novi Sad, Faculty Science in 1973, Master degree (1985) and Ph. D. degree (1992) booth in Robotics from the University of Novi Sad, Faculty of Technical Sciences. Since 1973 till 1980 she was

with the Faculty of Science in Novi Sad, and since 1980 she is with the Faculty of Technical Sciences, University of Novi Sad. Her research includes the fields of artificial intelligence, intelligent document management, and the applications of formal modeling in education. She can be contacted at: ftn_zora@uns.ac.rs

Milan Vidaković is holding the associate professor position at the Faculty of Technical Sciences, Novi Sad, Serbia. He received his PhD degree (2003) in Computer Science from the University of Novi Sad, Faculty of Technical Sciences. Since 1998 he has been with the Faculty of Technical Sciences in Novi Sad. Mr. Vidaković participated in several science projects. He published more than 60 scientific and professional papers. His main research interests include web and internet programming, distributed computing, software agents, and language internationalisation and localisation. He can be contacted at: minja@uns.ac.rs

Received: August 27, 2010; Accepted: April 5, 2011.

DOI: 10.2298/CSIS100804016L

Feature Diagram Formalization Based on Directed Hypergraphs

Miguel A. Laguna[1], José M. Marqués[1], and Guillermo Rodríguez-Cano[2]

[1] Department of Computer Science, University of Valladolid
Campus M. Delibes,
47011 Valladolid, Spain
{mlaguna,jmmc}@infor.uva.es
[2] wileeam@acm.org

Abstract. Feature models are used to represent the variability and commonality of software product lines (SPL), and to decide on the configuration of specific applications. Several variants based on tree or graph hierarchical structures have been proposed. These structures are completed with additional constraints, generally expressed in parallel with the feature diagram. This paper proposes the use of hypergraphs to integrate both concepts in a unique characterization. Therefore, the definition, validation and selection of feature configurations can be internally based on the hypergraph properties and well-known algorithms, while the concrete visual syntax remains unchanged for domain engineers. The implemented hypergraph algorithms have been tested using a complete set of feature diagrams. Finally a feature meta-model can be derived directly from the formal definitions, providing the foundations for building feature modeling tools.

Keywords: feature diagram, hypergraph, feature diagram configuration.

1. Introduction

Software product lines (SPL) constitute a successful reuse paradigm in industrial environments despite their complexity [3]. Feature diagrams (FD) represent the variability and commonality of software product lines and permit the configuration of each specific application to be selected.

Feature diagrams were first introduced in the Feature Oriented Domain Analysis (FODA) method [11], and Fig. 1 shows an example of a partial feature diagram of an eCommerce SPL inspired by Lau [15]. This proposal defined features as the nodes of an and/or tree related by various types of edges. As depicted in Fig. 1 a feature diagram is a structure diagram showing the hierarchical (parent/child) structure between the features. A feature diagram is composed of the tree root or concept (*eCommerce*), and its subfeatures showing mandatory, e.g. (*Payment*), optional, e.g. (*Registration*), and alternative features (*CreditCard, DebitCard, ElectronicCheque*). The relations (edges) between features (nodes) can be: AND, XOR and OPTIONAL relationships. To improve their expressiveness, several extensions have been proposed, incorporating the OR

decomposition [12], changing the visual syntax, or using directed acyclic graphs (DAG) instead of simple trees and UML-like multiplicity constraints to annotate multiplicities for sets of features [18]. Schobbens et al. [20] have surveyed and evaluated the diverse FD variants, clarifying the differences and establishing a generic semantics. The study classifies the existing proposals using several characteristics: the FD is a tree or a DAG, the constraints are textually or graphically shown, and the way the decomposition relationships (AND, XOR, OR, multiplicity) are expressed. They propose a generic formalization of the syntax and semantics of FD and a new non-redundant variant FD or VFD. One of the conclusions is that these FD description languages have a different concrete syntax but share many aspects of an abstract syntax and semantics based on graphs.

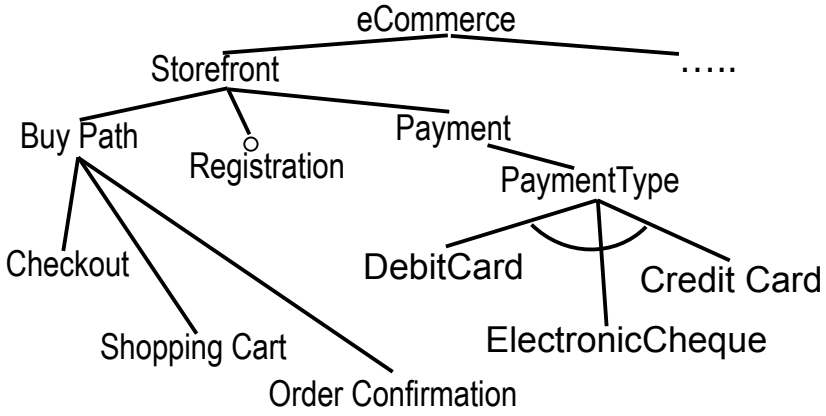


Fig. 1. Example of a FODA feature diagram

However, the use of standard graphs as the underlying structure of FDs forces the use of *ad hoc* internal representations to differentiate the diverse feature relationships. For example, the internal representation of the grouped alternative decomposition of the feature *PaymentType* of Fig. 1 (One-To-Many relationship), and the internal representation of the decomposition of the feature *Buy Path* (three binary relationships) must be different.

Moreover, an FD alone cannot capture all domain restrictions. Constraints between features (a feature requires another feature or two features are mutually exclusive) have been added, in textual or graphical formats, to complete the semantics of the models. If textual, separate constraint documentation must be handled to have the global picture of the model. Modifying a feature diagram can be difficult, since changing the hierarchy can cause modifications in constraints.

This leads us to consider the formalization of feature diagrams using directed hypergraphs [9]. A directed hypergraph is a generalization of the concept of directed graphs, an example is illustrated in Fig. 2 . In directed hypergraphs, One-To-Many, e.g. E_4 hyperarc, Many-To-One, e.g. E_2 hyperarc, and Many-To-Many,e.g. E_1 hyperarc, relations are naturally represented, including, as particular cases, labeled graphs, And-Or graphs, DAGs, and simple trees. Thus, conversion from a concrete FD to a hypergraph is simple and straightforward: features are the nodes of the hypergraph; decompositions and constraints are represented by hyperarcs.

The formalization of feature models using the structure of a hypergraph provides several benefits. Firstly, the use of hypergraphs as the underlying structure clarifies and simplifies the feature meta-model (only two basic types of elements are defined: features and relationships). Consequently, the definition and construction of automated tools is easier than in the case of graph based structures, as fewer elements are needed to express the FD semantics. More interestingly, analysis and configuration problems can be treated algorithmically, taking advantage of the progress made in hypergraph theory. From the practitioners' point of view, this will not affect them directly, because the FD language (i.e., the concrete syntax) is not modified.

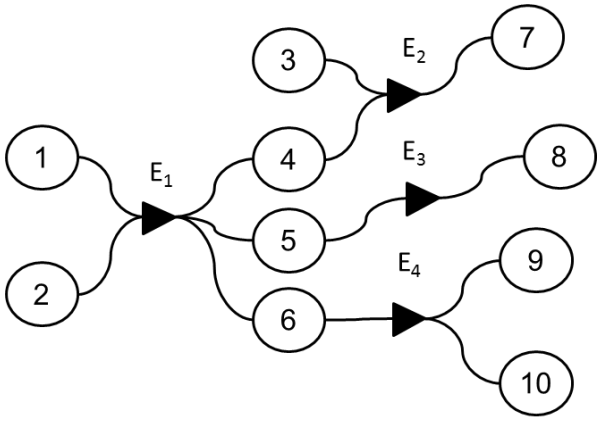


Fig. 2. An example of hypergraph

The rest of the article is as follows: the next Section introduces hypergraphs and formally defines the structure underlying an FD. Section 3 analyzes the configuration problem and sketches the hypergraph algorithms used for deriving the configuration, starting from a set of features selected by the user. Experimental results are also provided. Section 4 uses the formal definition to build a feature meta-model. Finally, Section 5 presents related work and Section 6 concludes the paper and considers future work.

2. Hypergraphs and Feature Diagrams

This section describes the conceptual framework and the formal structure, based on hypergraphs, used to model features and relationships among features in an FD. First, we define the feature modeling concepts to be used in the rest of the paper. Next, basic definitions of hypergraphs are given. Finally, the formulation of FDs is presented.

In general terms, a feature model defines features and their dependencies, covering the commonality and variability of software products in a Software Product Line. An FD is a structure defined by features, decomposition relationships, and constraint relationships. To model an FD by means of a supporting hypergraph we are only interested in its underlying structure formed by the following elements:

Features The collection of identified characteristics of a system which determines the scope of the Product Line under study. A feature is "a system property that is relevant to some stakeholder and is used to capture commonalities or discriminate between systems" [5].

Root Feature Also known as the concept, this is a distinguishable feature used to denote the feature diagram. It is always present in the feature model.

Feature relationships These are used to model directed associations between features. They are used to model the decomposition of a feature (parent feature) into one or more detailed features (children features). Each feature relationship will be labeled with a UML-like multiplicity value specifying the range of allowable features to be selected.

Constraints To further restrain the valid configurations of an FD, additional constraints can be formulated as a (dependency) relationship between features. The constraints considered are *requires* and *mutex*. Requires establishes a compulsory relationship between a source feature and a subset of target or required features. The *requires* relationship defines an implication where, if the source feature is selected, then the implied set of features must be selected as well. Unlike requires, the *mutex* relationship defines an exclusion between features. Thus, if two or more features are associated by a *mutex* constraint, one and only one of the features can be selected in a product configuration.

A hypergraph is a generalization of a graph wherein edges can connect more than two vertices and are called hyperedges. Directed hypergraphs extend directed graphs and have been used as a modeling and algorithmic tool in many areas: formal languages, relational databases, manufacturing systems, public transportation systems, etc. A technical, as well as historical, introduction to directed hypergraphs has been given by Gallo et al. [9]. The main reason for introducing this type of graph is to represent Many-to-Many relations, for which simple DAG or trees are not well equipped. Definitions of *hypergraph* and the *Backward Star* function, useful for our purposes, are given:

Definition 1. *[8] A directed hypergraph, or simply hypergraph, is a pair* $H = (V, E)$*, where* V *is a non empty set of vertices (or nodes) and* E *is a set of*

directed hyperedges; a directed hyperedge or hyperarc $e \in E$ is an ordered pair, $e = (T(e), H(e))$, where $T(e) \subseteq V$ is the tail of e, while $H(e) \subseteq V \backslash T(e)$ is its head.

Definition 2. *[8] Given a hypergraph $H = (V, E)$ and a node $v \in V$, Backward Star of node v, $BS(v) \subseteq E$, denotes the set of hyperarcs entering v, $BS(v) = \{e \in E \mid v \in H(e)\}$*

The hypergraphs we use in the rest of the article are F-hypergraphs, a particular case of acyclic directed hypergraphs, characterized by the fact that the tail cardinality of the hyperarcs is always one (Forward hyperarcs, or simply F-hyperarcs). So the next definition states:

Definition 3. *An F-hypergraph is a pair $H = (V, E)$, where V is a non empty set of nodes and E is a set of F-hyperarcs; an F-hyperarc $e \in E$ is an ordered pair, $e = (t(e), H(e))$, where $t(e) \in V$ is the tail of e, while $H(e) \subseteq V - \{t(e)\}$ is its head.*

2.1. Feature Diagrams as Hypergraphs

A feature model, as the result of a Domain Analysis, defines features and their relationships, the feature diagram being its key element. Therefore, the starting point for the formulation of an FD in terms of hypergraphs is to associate one node with each feature and one hyperarc with each feature relationship. The feature diagram can thus be described by an F-hypergraph. Each hyperarc is assigned a label which corresponds to the multiplicity of the relationship.

More formally, for a given FD, let F denote the set of all its features, $F = \{f_1, f_2, \ldots, f_k\} \cup \{f_c\}$, where f_c is the concept or root feature of the FD; f_c is the only node not contained in the head of any hyperarc of the hypergraph, i.e. it is the only node in the FD whose Backward Star, $BS(f_c)$, is the empty set.

A feature relationship takes the general form $e = (f, S, [m \ldots n])$, where:

- f is a feature of F
- S is a non empty subset of features of $F - \{f, f_c\}$, $|S| \geq 1$, and
- $[m \ldots n]$ is the relationship multiplicity with m, n integers, $m \geq 0; 0 < n \leq |S|; m \leq n$

The semantics of the relationship multiplicity is that if feature f is selected, at least m and no more than n features of subset S must be selected. From the point of view of feature modeling, f is the parent feature and S the set of its children features or subfeatures. We associate a hyperarc with each feature relationship e, having f as its tail and S as its head. Fig. 3 shows how feature decomposition relationships of an FD language are formulated as hyperacs.

The constraint relationships will be formulated as feature relationships. The *requires* relationship establishes, in general, a compulsory relationship between a feature f and a subset of features R such that, if f is selected, all features of subset R must be selected. In other words, a *requires* relationship can be

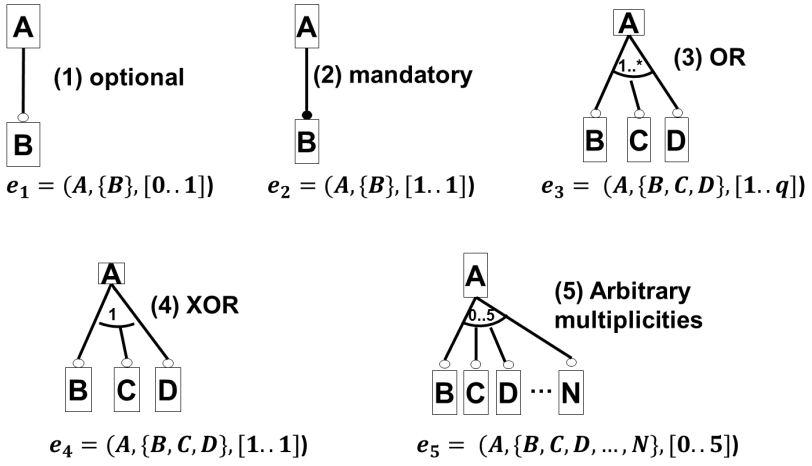


Fig. 3. Example of feature relationships [18] modeled as hyperarcs

written in terms of a feature relationship as $e = (f, R, [q \ldots q])$ with $|R| = q$. We would like to point out that introducing this kind of hyperarc, cycles could arise in the hypergraph. This is an undesired and nonsensical situation; therefore, once all the *requires* hyperarcs have been defined; the acyclicity of the hypergraph could be tested with the F-Acyclic procedure described in [8].

The transformation of a *mutex* relationship into a feature relationship is slightly more elaborate. It must be reinterpreted as a feature relationship from a common and always present feature – the root feature fc – to the involved features in a *mutex* constraint. In brief, if a *mutex* constraint exists among a subset of features of FD, $M \subset F - \{fc\}$, then the constraint can be modeled as: $e = (f_c, M, [0..1])$

Some authors propose enhancing *requires* and *mutex* binary constraints with arbitrary n-ary logical expressions between features. The semantics of these complex expressions can be difficult to understand but, in any case, they can be added as hyperarcs to the hypergraph [17].

We now have a one-to-one correspondence between the concepts of a feature model and the structure of a hypergraph. It should be pointed out that the above formalization of features and feature relationships belongs to the abstract syntax level being applicable, without loss of information, to the concrete syntax variants of most representative FD languages [20].

Figure 4 depicts a feature diagram of a simple *WebPortal* system based on the feature model described by Mendonça et al. [17]. As pointed out above, there are some semantics which can not be expressed in a feature diagram using features and decomposition relationships, therefore additional constraints are required, the two best known being *requires* and *mutex*. Feature constraints

to be considered for the *WebPortal* system displayed in figure Fig. 4 , expressed in textual form, are:

WebPortal constraints
$Auth.$ **REQUIRES** $UserLogin$
$Transfer$ **REQUIRES** $https$
Ms **MUTEX** $https$

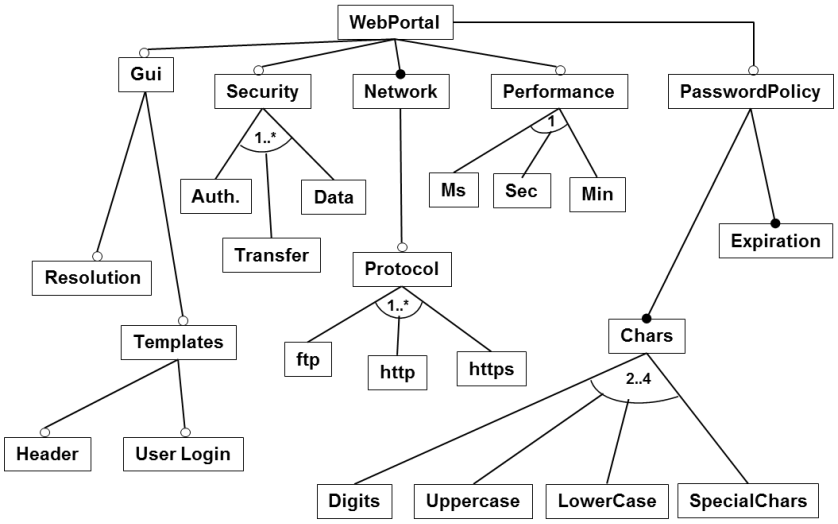


Fig. 4. *WebPortal* system feature diagram inspired by [17] using Riebish et al. [18] notation

To describe the *WebPortal* feature model as an f-hypergraph model, the aforementioned mapping rules will be applied. Let $WP = (F, E)$ denote the derived hypergraph, where F is the set of features, the nouns shown in Fig. 4, $F = \{Gui, Security, ...\} \cup \{WebPortal\}$ and E are, repectively, the set of hyperarcs representing the decomposition and the constraint relationships between features. These transformation are shown in table 1, where $n_1 \rightarrow n_2$ means n_1 is a parent of n_2; n_2 is a subfeature of n_1; and $n_1 \rightarrow [n_2, n_3, \ldots, n_k]$ means n_1 is a parent of the set of features $n_2, n_3, \ldots, n_k$, and each $n_i \; i = 2 \ldots k$ is a subfeature of n_1

Figure 5 illustrates the full F-Hypergraph transformation of the *WebPortal* system feature diagram depicted in Fig. 4 according to the proposed hyperarc transformation processes shown in Table 1. Note that the hyperarcs representing *require* and *mutex* constraints between features have been identified using an empty arrowhead instead of the solid arrowhead used for hyperarcs derived from the FD decomposition relationships. The existence of two kinds of adorn-

Table 1. F-arcs of F-hypergraph $WP = (F, E)$

FEATURE MODEL RELATIONSHIP	F-ARC
WebPortal → Gui	$(WebPortal, \{Gui\}, [0 \dots 1])$
WebPortal → Security	$(WebPortal, \{Security\}, [0 \dots 1])$
WebPortal → Network	$(WebPortal, \{Network\}, [1 \dots 1])$
WebPortal → Performance	$(WebPortal, \{Performance\}, [0 \dots 1])$
WebPortal → PasswordPolicy	$(WebPortal, \{PasswordPolicy\}, [0 \dots 1])$
Gui → Resolution	$(Gui, \{Resolution\}, [0 \dots 1])$
Gui → Templates	$(Gui, \{Templates\}, [0 \dots 1])$
Security → [Auth., Transfer, Data]	$(Security, \{Auth, Transfer, Data\}, [1 \dots 3])$
Network → Protocol	$(Network, \{Protocol\}, [0 \dots 1])$
Performance → [Ms, Sec, Min]	$(Performance, \{Ms, Sec, Min\}, [1 \dots 1])$
PasswordPolicy → Expiration	$(PasswordPolicy, \{Expiration\}, [1 \dots 1])$
PasswordPolicy → Chars	$(PasswordPolicy, \{Chars\}, [1 \dots 1])$
Templates → Header	$(Templates, \{Header\}, [0 \dots 1])$
Templates → User Login	$(Templates, \{UserLogin\}, [0 \dots 1])$
Protocol → [ftp — http — https]	$(Protocol, \{ftp, http, https\}, [1 \dots 3])$
Chars → [Digits, Uppercase, LowerCase, SpecialChars]	$(Chars, \{Digits, Uppercase, LowerCase, SpecialChars\}, [2 \dots 4])$
$Auth.$ **REQUIRES** $UserLogin$	$(Auth, \{UserLogin\}, [1 \dots 1])$
$Transfer$ **REQUIRES** $https$	$(Transfer, \{https\}, [1 \dots 1])$
Ms **MUTEX** $https$	$(WebPortal, \{Ms, https\}, [0 \dots 1])$

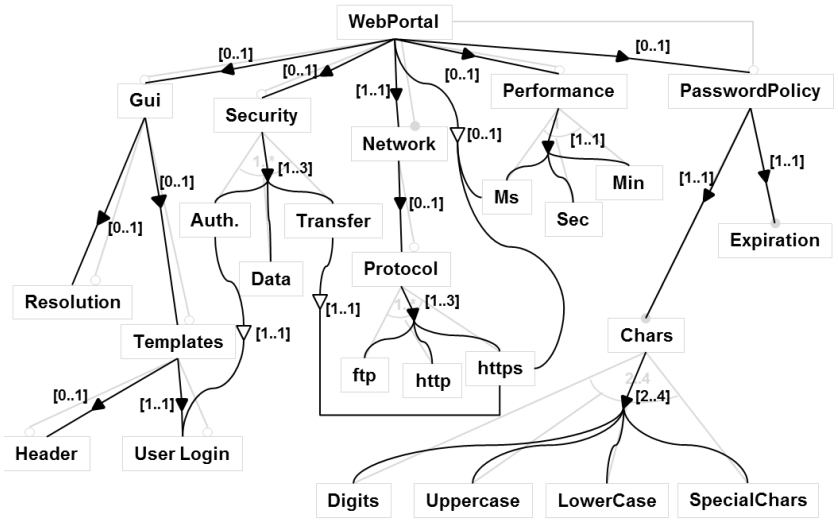


Fig. 5. *WebPortal* system F-hypergraph transformation

ments for hyperarcs has no special meaning, being used only for graphical purposes.

2.2. Formal Definitions

This subsection summarizes the described concepts in a compact form:

Definition 4. *A multiplicity value denoted as* $[m \dots n]$ *is a pair of integers* (m, n) *with* $m \geq 0$, $n > 0$ *and* $m \geq n$.

Definition 5. *An Unconstrained Feature Diagram is an (arc labeled) acyclic F-hypergraph* $UD = (F, E, f_c)$ *where:*

- F *is its set of nodes (or features)*
- $E = \{e_1, e_2, \dots, e_p\}$ *is the set of decomposition F-arcs, with* $e_i = (f_i, R_i, [m \dots n])$, $R_i \subseteq F - \{f_i\}$ *and* $n = |head(e_i)|$ *for* $i = 1, \dots, p$
- $BS(f_c) = \emptyset \ \wedge \ BS(f) \neq \emptyset \ \ \forall f \in F - \{fc\}$

A particular type of Unconstrained Feature Diagram is the Feature Tree. If each feature has no more than one parent, then the structure is a hypertree:

Definition 6. *A Feature Tree FT is an Unconstrained Feature Diagram, such that each feature has at most one entering hyperarc (root* f_c *has none):* $|BS(f)| = 1 \ \forall f \in F - \{fc\}$

Constraints are introduced as extensions, adding additional hyperarcs with multiplicity $1 \dots n$ (*requires*) or $0 \dots 1$ (*mutex*)

Definition 7. *Given an Unconstrained Feature Diagram* $UD = (F, E, f_c)$*, a Constrained Feature Diagram, or simply a Feature Diagram, is an acyclic F-hypergraph* $FD = (F, E', f_c)$*, where*

- $E' = E \cup E_r \cup E_t$
- $E \cap E_r = E \cap E_t = E_r \cap E_t = \emptyset$
- $E_r = \{r_1, r_2, \dots, r_k\}$ *is the set of requires constraints F-arcs, with* $r_i = (f_i, R_i, [q \dots q])$ *where* $f_i \in F$, $R_i \subset F - \{f_i\}$ *for* $i = 1, \dots, k$ *such that* $1 \leq |head(r_i)| = q$
- $E_t = \{t_1, t_2, \dots, t_l\}$ *is the set of mutex constraints F-arcs, with* $t_i = (f_c, R_i, [0 \dots 1])$*, where* $R_i \subset F - \{f_c\}$ *for* $i = 1, \dots, l$ *and* $|head(t_i)| \geq 2$
- *For each* $e \in E'$*, the function isConstraint is defined:*
  - *isConstraint(e) = true, if* $e \in E_r \cup E_t$
  - *isConstraint(e) = false, if* $e \in E$

The last Boolean function is defined so as to be used in the configuration process, as explained further in Section 3.

As it is the most complete type of FD, from here on, we refer to the Constrained variant simply as the Feature Diagram or FD and denote it by default as $FD = (F, E, f_c)$. Section 3 shows how the well-known hypergraph traversal algorithm can contribute to the FD validation and configuration procedures.

3. Configuration of Feature Diagrams

A (partial) configuration of a Feature Diagram is a sub-set of the original Feature Diagram where the variability is (partially) removed. In general, a manual process of feature selection is carried out, obeying the constraints expressed in the diagram. Some of these constraints are implicitly imposed by the diagram structure. Defining *mandatory* (*non-mandatory*) as decompositions where the minimum multiplicity is equal to (less than) the number of children, the following rules apply:

Rule 1 The root feature must be present in any configuration.
Rule 2 A feature can be selected only if at least one of its parents is selected.
Rule 3 If a feature is present, the features connected to it, through mandatory decompositions, must be selected.
Rule 4 If a feature is present, the number k of non mandatory features selected as children of its decompositions must be between the minimum and maximum of the original hyperarc multiplicity: $m \leq k \leq n$.

Two more rules are imposed by the *requires* and *mutex* constraints:

Rule 5 *Requires* constraints mean that, for each feature in the configuration, all the elements required by it must also be present. In the hypergraph representation, this is equivalent to a mandatory decomposition (Rule 3).
Rule 6 *Mutex* constraints over a set of features mean that, if an involved feature is present in the configuration, the others must be absent. In the hypergraph representation, this is equivalent to a non-mandatory decomposition with multiplicity equal to [0..1] (Rule 4).

Consequently, the configuration procedure can be applied uniformly to the constrained hypergraph, instead of dividing it into two problems or transforming the feature tree (or graph) into a set of propositional formulas, as proposed in the literature [17]. In fact, generalizing the several variants, all the above rules can be reformulated in a comprehensive way: In a valid configuration, defined as a subset of features of one FD, the present features must satisfy two simple properties:

Property 1 For each feature other than the root, at least one of its *structural* parents (the tail of a hyperarc not representing a mutex/requires constraint) must also be present.
Property 2 For each leaving hyperarc of the considered feature, with $[m \dots n]$ multiplicity, at least m and at most n children features of the head of the hyperarc must also be present in the configuration

These properties are enough to accomplish the rules 1 to 6. Considering the possible combinations of children given by the hyperarc multiplicity, the remaining rules can be trivially deduced. The second property implicitly defines a type of reduction [8] of the original FD hyperarcs into each one of the possible configuration hyperarcs.

Definition 8. *Given a Feature Diagram $FD = (F, E, f_c)$, and a hyperarc $e \in E$ where $e = (tail(e), head(e), [m \dots n])$, we define a C-reduction of e as a hyperarc $e_G = (tail(e_G), head(e_G))$ such that $tail(e_G) = tail(e)$, $head(e_G) \subseteq head(e)$, $m \leq |head(e_G)| \leq n$.*

Definition 9. *A Valid Configuration $G = (F_G, E_G)$ is a hypergraph obtained from a (Constrained) Feature Diagram $FD = (F, E, f_c)$ by replacing each selected hyperarc of E by one of its C-reductions:*

- *F_G is a subset of features of F: $F_G \subseteq F$*
- *E_G is a set of hyperarcs $E_G = \{e_G\}$ where e_G is a C-reduction of one $e \in E$.*
- *The root is present: $f_c \in F_G$*
- *For each feature in the configuration, at least one of its parents in FD is present, giving as a result an F-connected hypergraph:*
  $\forall f \in F_G \; f \neq f_c \; \exists e \in E \wedge isConstraint(e) = false \wedge f \in head(e) \wedge tail(e) \in F_G$
- *For each feature f in F_G, for each leaving hyperarc e of f in E, $e = (tail(e), head(e), [m \dots n])$, with $m > 0$, one C-reduction e_G of e, must be present in the configuration:*
  $\forall f \in F_G \wedge e \in E \wedge tail(e) = f \wedge m > 0 \Rightarrow \exists e_G \in E_G$ such that e_G is a C-reduction of e

Considering that the semantics of feature modeling is expressed by the concept of FD configuration [20], we can say that an FD is valid if at least one valid configuration can be derived from it and if each feature of the FD is present in at least one configuration (no *dead* features). The trivial cases are one FD with only one feature (the root itself) or with only mandatory features (no variability at all, only one valid configuration). For the useful cases, to validate one FD, it is enough to prove that each feature is present in at least one valid configuration. The next Subsection gives a procedure to find a valid configuration of one FD, given a set of selected features. The repeated application of that procedure to each individual feature of the FD will serve to trivially prove the validity of the FD itself.

3.1. Configuration Procedure of a Feature Diagram

The given definition will guide the configuration process. Once the application engineer has expressed his/her preferences, by selecting a set of features, and has checked their compatibility (i.e., there are no mutex or multiplicity conflicts between them), we can find a usual problem: it is possible that feature decompositions with no children selected remain undefined (hyperarcs with minimum multiplicity m, $0 < m < |head(e)|$). There are at least two ways in which the configuration procedure can be dealt with: a) finding the (probably ordered) set of all valid configurations that fulfill the defined selection; and b) guiding the engineer until a unique valid configuration is found. The first option is a complete but computationally costly solution. The second is more realistic, but it remains a largely manual process, accomplished with the help of FD modeling

tools. Staged configuration [7] is a classical approach for solving this problem in several steps.

To facilitate the configuration process in these undefined hyperarcs, we think that it is useful to predetermine a topological order in the set of features included in the head of each hyperarc. This option implies that the domain engineer has assigned a preference order to each group of features (alternatively, the "weight" of the feature plus its mandatory descendants could be automatically calculated and assigned to the features [8]). The aim is to have a (set of) default feature(s) when there is no explicit decision. An example can clarify the idea: in an e-commerce product line, credit card payment is more frequent than check or transfer based payments and, in consequence, if the application engineer does not explicitly decide to change the payment method, credit payment will be selected by default.

In any case, the possible configurations can be generated in two steps that try to accomplish properties 1 and 2 respectively:

Step 1 The partial configurations that include the selected features are found. This step is deterministic in the sense that one path from each selected feature to root must be included, and (recursively) mandatory descendants of each feature in the configuration must be added. If valid (no multiplicity limits are violated), the resulting partial configurations can be communicated to the engineer or used in step 2.

Step 2 Each resulting partial configuration is completed using a second procedure that finds the most "economical" configuration of the FD. For each undefined hyperarc (the number of selected features of its head is less than the minimum multiplicity and the hyperarc tail is present) the default feature descendant(s) are added to reach the minimum multiplicity. Again, mandatory descendants of each new feature are added and constraints are checked.

The first step is a variation of the hypergraph $BVisist(r, H)$ algorithm. The second one can be designed as a refinement of the $FVisit(r, H)$ algorithm, both described and analyzed in [8], and running in $\mathcal{O}$(size(H)) time, H being the traversed hypergraph. Thus, given a (Constrained) Feature Diagram $FD = (F, E, f_c)$, and U being an identified (selected manually) subset of features of F: $U \subset F$, a set of valid configurations $G_i = (F_{G_i}, E_{G_i})$ is obtained in two steps.

3.2. Step 1 Implementation

We say that one hyperarc $e \in F$ is a *parent* hyperarc of the feature u if the head of the hyperarc contains u, $u \in head(e)$, and it is *structural* (i.e., not a constraint type hyperarc). The value *used(e)* gives the number of features of *head(e)* included in the configuration so far. For each present feature, at least one parent hyperarc must be included in the configuration. We always start from an initial configuration of FD, $G_0 = (F_0, E_0)$ where F_0 includes only the root feature of FD and E_0 has zero selected hyperarcs.

For U and G, where G is a (partial) configuration of FD, a procedure *configure(U,G)*, adapted from $BVisist(r, H)$ [8] is applied (See Procedure 1 basic scheme). For each feature $u \in U$, u is selected and removed. For each hyperarc e, *parent* of u, (i.e., *requires* or *mutex* arcs are not selected), and if it is a *valid parent hyperarc* (the hyperarc $used(e)$ value is less than the hyperarc maximum multiplicity), the configuration process continues: the tail of e is added to the list of selected features and a new recursive execution of procedure 1 is launched, starting from the current configuration. If there are no *valid parent hyperarcs* for a feature, the procedure execution aborts (though other branches can continue).

Clearly the original complexity of the procedure increases with the number of features with more than one structural parent, being multiplied by each number of structural entering arcs a_i of each feature f_i except the root, i. e. $\mathcal{O}(size(H).\Pi a_i)$ (though all or most of the values are one in the FDs published in the literature). In the implemented version of the procedure, each time one feature is added, two operations are used for efficiency reasons: (1) its mandatory descendants are also added to the selected features, using an auxiliary procedure; (2) the existence of conflicts in the resulting partial configuration is tested using an auxiliary $compatible(G)$ function (returns false if $used(e) > n$ in any hyperarc of the configuration) to discard illegal configurations as soon as possible. Procedure 1 transforms the initial U and $G_0 = (F_0, E_0)$ into one set of partial configurations $G'_i = (F_{G_i}, E_{G_i})$, all compatible with the expressed requirements.

For each resulting modified G'_i, property 1 holds at this point, that is, each feature has at least one structural parent (and no conflicts are present). Property 2 must be accomplished in a second step.

3.3. Step 2 Implementation

The procedure $complete(G'_i)$ inspired by $FVisist(r, H)$ [8], is applied to each modified partial configuration G'_i. The original $FVisist(r, H)$ finds all nodes connected to root r and returns a set of paths connecting them to r. This must be adapted in order to limit the number of features of $head(e)$ to be examined. Note that, for each undefined hyperarc, only m features must be selected, $[m \dots n]$ being its multiplicity value. The algorithm can be improved if we consider the default features first. Thus, if valid, the algorithm will reach the configuration where all the choices are the default features(s). Only when that configuration is illegal, is backtracking applied and the algorithm continues to search for a valid configuration $G_i = (F_{G_i}, E_{G_i})$.

As in procedure $configure(U, G)$, in the optimized implemented version, the mandatory descendants of each new feature selected are also added to the selected features, using an auxiliary procedure (not shown in the basic scheme of procedure $complete(G_i)$). If new feature(s) with no parents are added to the configuration, a new call to the $configure(U, G)$ procedure is necessary to find their parents before completion (Uincludes the new features without parents).

Procedure configure(U,G)

```
Generation of the staged (partial) configurations;
while U ≠ ∅ do
    // Select and remove node u ∈ U
    u ← first(U) ;
    U ← U − {u} ;
    // Mark node u ∈ G as selected
    mark(u,G) ;
    // Valid number of parents
    parents ← 0 ;
    // Try entering non-constrained arcs
    foreach e ∈ BS(u) and isConstraint = false do
        // Select hyper-arc when possible
        if used(e) < max(e) then
            parents ← parents + 1 ;
            used(e) ← used(e) + 1 ;
            if not getMark(tail(e), G) then
                mark(tail(e), G) ;
            end
            U ← U ∪ tail(e) ;
            configure(U,G) ;
        end
    end
end
// Finished path
G ← void ;
// Check for non-valid configurations
if parents = 0 then
    return ;
end
```

Again the original complexity of the algorithm is increased by the number of possibilities that can be chosen in each hyperarc with $0 < m < |head(e)|$ (2 in [1..1] and $|head(e)| = 2$, 3 in [1..1] and $|head(e)| = 3$, 3 in [2..2] $|head(e)| = 3$). This combinatorial explosion is alleviated because, in practice, we only select the first *m* features of the ordered set of children features and we only intend to find the first valid configuration.

The minimal set of features present in a configuration if no features are selected (built applying both procedures to the root feature and selecting the first *m* children in each hyperarc) constitutes the *default* feature configuration. The minimal set of features present in all the possible configurations (built applying only configure to the root feature and selecting only *m* mandatory/ requires features where $m = |head(e)|$ in each considered hyperarc) constitutes the *core* features. The base package of the SPL architecture [13] is a design solution to these core features.

3.4. Empirical Evaluation

An implementation in Java has been coded to test the procedures and to estimate the time needed to reach the partial and final configurations. The FDs used for experimentation were downloaded from the SPLOT[3] project [16]. The configurations were created in a quasi-random way (if an alternative feature is included, the rest of the features of the group are discarded to trivially avoid inconsistent configurations). The number of selected features over the total (RCS in the Tables) range between 1% and 35%. The real FDs of the SPLOT Web (all of which are originally trees with constraints) were divided for their study into several groups according to size.

The results corresponding to the biggest real models are shown in Table 1. The columns contain the FD Size (FD_S), Extra constraints representativeness (ECR), Relative configuration initial size (RCS) , Zero (mandatory descendants of root), First, and Second Algorithm Average time in ms (S_0, S_1, S_2), Configuration time (S_{1-2}), First or second staged-configuration size (average ratio of the number of valid configurations obtained in first, C1, or second, C2, algorithm execution). The dashes mean that no valid random configuration is generated, as the maximum number of features to be chosen was exceeded. It can be seen that, in general, when the number of initially selected features (RCS column) is increased, the time of algorithm 1 grows while the time of algorithm 2 decreases. The total sum remains acceptable (less than 0.4 seconds). The computer used was a Mac (OS X), equipped with an Intel processor (Core 2 Duo 2.16 Gigahertz), 2 Gb of memory.

A second group of studied FDs corresponds to *benchmark* models, with a considerably bigger size: up to 10,000 features and 1,000 constraints. Some constraints (in particular ternary CNF expressions) had to be previously transformed into hypergraph constructions and/or binary constraints to be dealt with,

[3] http://splot-research.org/

Procedure complete(G_i)

Generation of a complete configuration from a partial configuration;

```
// Collect hyperarcs to be completed
foreach e ∈ E∧ getMark(tail(e),G_i) do
    W ← W ∪ {e} ;
end
while W ≠ ∅ do
    // Select and remove node w ∈ W
    w ← first(W ) ;
    W ← W − {w} ;
    // Try hyper-arc's head non-selected features
    foreach f ∈ head(w) do
        // Select feature f and check for no conflicts
        if used(w ) < min(w) and not getMark(f,G_i) then
            mark(f,G_i) ;
            used(w ) ← used(w ) + 1 ;
            if compatible(G_i) then
                // Find partial configurations for require
                   constraints
                if count(getRequire(G_i)) > 0 then
                    G_iv ← configure(getRequire(G_i),G_i) ;
                else
                    G_iv ← G_i;
                end
                // Complete derived partial configurations or
                   current
                foreach g ∈ G_iv do
                    G_i' ← complete(g) ;
                    if compatible(G_i') then
                        G_i ← G_i ∪{ G_i' } ;
                    else
                        // Configuration no valid
                        G_i ← void ;
                    end
                end
            else
                // Configuration no valid
                G_i ← void ;
            end
        end
    end
end
if W = ∅ then
    // Configuration is completed
end
```

as described in [19]. The elapsed time in these cases is longer but acceptable with a peak of 18.5 seconds (Table 2).

Table 2. Time (ms) of execution of the configuration algorithms applied to real FDs

				Configuration					
Feature model	FD_s	ECR(%)	RCS	S_0	S_1	S_2	$S_{[1-2]*}$	C_1	C_2
Home			5 %	4.36	1.98	12.66	15.03	1.0	1.0
Integration	67	11.9	15 %	8.72	4.19	7.78	12.54	1.0	1.0
System			25 %	14.59	1.37	3.17	5.29	1.	1.0
			35 %	-	-	-	-	-	-
Ecological			5 %	9.04	3.52	25.59	29.65	0.72	0.72
car	94	4.3	15 %	19.71	14.13	0.0	14.2	0.0	0.0
			25 %	28.13	12.41	0.0	12.5	0.0	0.0
			35 %	-	-	-	-	-	-
Electronic			5 %	39.79	28.58	287.98	317.93	1.0	0.84
shopping	287	11.8	15 %	77.1	55.29	188.74	245.71	1.0	0.8
			25 %	124.2	79.46	147.8	229.73	1.0	0.8
			35 %	199.17	110.97	106.54	219.71	1.0	0.6

As all these models are based on trees, we use a modified version of the "Electronic shopping" FD (the last row of table 1) to test the influence of features with more than one parent in the performance of the first and second step configuration algorithms (the first algorithm creates several alternative partial configurations, completed by the second algorithm). Table 3 shows the results. Although the time increases notably, the relation time/number of found configurations is of the same order of magnitude. Thus, we can conclude that the use of hypergraphs as a practical tool with regular FDs is viable. More details of the conducted tests can be found in [19].

4. Feature Meta-Model

One of the advantages of the definition of Feature Diagrams as F-hypergraphs is that we have only two types of elements, features and relationships, instead of introducing an additional element (grouped features) to complete the semantics. In consequence, the definition and implementation (as CASE tools) of the meta-model is easier. The proposal is modular, allowing several versions, from the simplest Tree based meta-model to the complete constrained F-hypergraph meta-model. More complete details of the meta-model definition and tool implementation with GMF can be found in [14]. The definition style uses the package merge mechanism and is the same that the UML 2 meta-model uses extensively in the OMG documentation. This approach allows all the variants of the

Table 3. Time (ms) of execution of the configuration algorithms applied to benchmarks FDs [16]

				Configuration					
Feature model	FD_s	ECR(%)	RCS	S_0	S_1	S_2	$S_{[1-2]*}$	C_1	C_2
SPLOT-			1 %	69.68	7.72	3544.49	3553.85	0.98	0.08
FM-50-	517	9.6	2 %	72.61	15.92	3643.55	3661.15	0.96	0.01
SAT-1			5 %	90.3	47.91	3218.32	3268.04	0.81	0.0
SPLOT-			1 %	71.16	5.71	3115.95	3123.34	0.99	0.34
FM-50-	511	9.4	2 %	71.48	12.78	3004.28	3018.68	0.95	0.25
SAT-2			5 %	88.83	41.88	2759.61	2803.5	0.82	0.16
SPLOT-			1 %	255.7	21.15	18456.9	18481.16	0.78	0.26
FM-100-	1034	9.6	2 %	274.52	60.65	9239.41	9301.71	0.36	0.5
SAT-1			5 %	329.89	224.79	245.76	470.83	0.01	0.0
SPLOT-			1 %	217.36	23.83	15786.21	15812.61	0.74	0.38
FM-100-	1036	9.4	2 %	227.58	56.24	5785.63	5842.91	0.25	0.06
SAT-2			5 %	283.67	184.34	0.0	184.56	0.0	0.0

Table 4. Time (ms) of execution of the configuration algorithms applied to customized FDs with multiple parent features

				Configuration					
Cust.	FD_s	ECR(%)	RCS	S_0	S_1	S_2	$S_{[1-2]*}$	C_1	C_2
$F_2 = 2$			5 %	44.33	57.26	346.2	405.2	1.36	1.26
$\wedge$	287	11.8	15 %	85.6	147.34	358.66	509.93	1.83	1.43
$F_3 = 0$			25 %	135.71	189.38	281.41	474.49	1.95	1.0
			35 %	209.7	205.67	216.59	426.25	1.9	0.9
$F_2 = 2$			5 %	47.45	59.96	384.5	447.08	1.36	1.3
$\wedge$	287	11.8	15 %	85.27	266.1	735.23	1006.87	3.57	2.83
$F_3 = 1$			25 %	140.35	470.29	881.43	1363.0	6.0	3.0
			35 %	229.68	559.04	829.98	1404.13	6.6	4.6
$F_2 = 2$			5 %	47.32	218.84	1654.22	1881.47	4.58	6.5
$\wedge$	287	11.8	15 %	95.34	1303.77	4469.7	5810.7	20.77	12.8
$F_3 = 3$			25 %	152.26	3240.44	8502.7	11836.78	45.95	26.45
			35 %	254.95	3762.53	10133.82	14053.74	57.6	43.2
$F_2 = 3$			5 %	42.65	216.88	1680.09	1903.33	4.56	5.7
$\wedge$	287	11.8	15 %	89.61	1473.03	5867.12	7380.1	22.73	18.8
$F_3 = 3$			25 %	151.83	3737.02	10166.15	14012.18	56.65	31.2
			35 %	251.19	8595.86	30783.49	39636.66	115.2	97.2

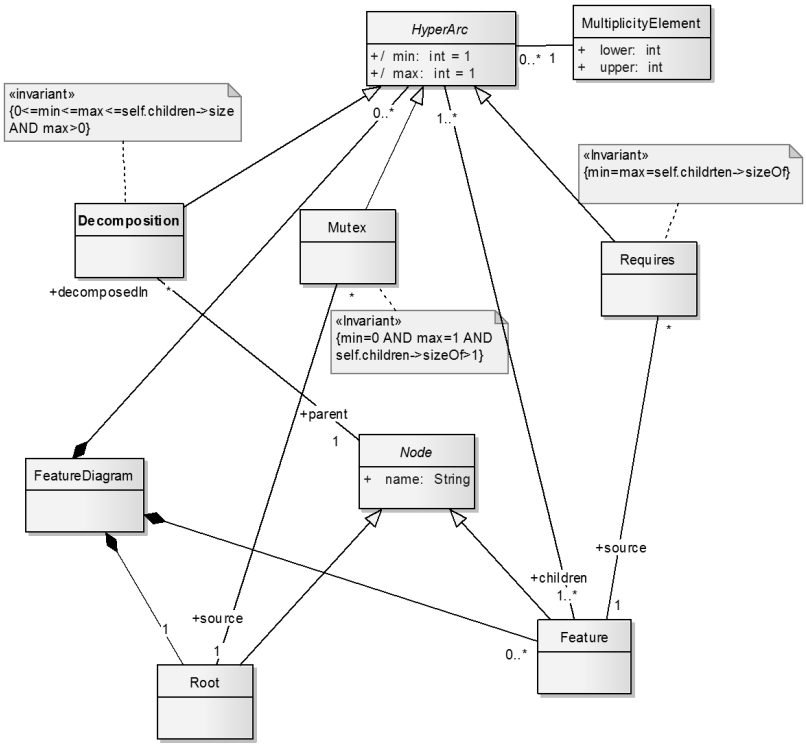


Fig. 6. Detail of the proposed Feature Meta-model [14]

feature diagrams mentioned in Section 2 to be covered. The details of the hypergraph based constrained meta-model are shown in Fig. 6. *HyperArc* and *Node* are abstract meta-classes. A *FeatureDiagram* has a *Root* (feature), a set of zero or more (non-root) *Features*, and a set of *Decompositions*. Each Decomposition connects a parent Node (Root or Feature) with one or more children Features. As multiplicity of children meta-association indicates, a Feature can be the child of more than one Decomposition and, indirectly, of a parent Feature. (If we change the multiplicity to 1..1, we convert the structure into a tree with a root that has no parents.) Decomposition has an associated *MultiplicityElement* that must conform to the associated OCL constraint: the maximum value (upper) must be less than or equal to the number of children of the Decomposition. Finally *Mutex* and *Requires* meta-classes are specializations of HyperArc with the adequate invariant to express the semantics of these constraints, as explained in Section 2.2: the fixed multiplicity of Mutex is 0..1, the multiplicity of requires is n..n (if n is the number of involved children, 1..1 being the typical situation). A basic implementation was presented in [14]. As part of our

industrial oriented work, we have previously implemented a Feature Modeling Tool (FMT)[4] as a *plug-in* for Microsoft Visual Studio IDE. The meta-model we used was based on constrained trees, validation was external, and the configuration used a staged approach. Work in progress includes the change of the internal meta-model and the incorporation of the validation and configuration algorithms.

5. Related Work

Starting with the original FODA proposal [11], several variants of feature diagrams have been proposed: FORM [12] is an extension where feature diagrams are single-rooted directed acyclic graphs (DAG) instead of simple trees. FeatureRSEB [10] also uses DAGs and changes the visual syntax, including a graphical representation for the constraints requires and mutex. Other authors, such as Czarnecki et al. [5,6] and Batory [1], continue to use trees as the main structure (however Czarnecki et al. add OR decomposition, graphical constraints, and distinguish between group and feature cardinalities). Riebisch et al. [18] replace AND, X-OR, and OR by multiplicities combined with mandatory and optional edges.

Cechticky et al. proposed a notation without solitary features in an attempt to reduce the number of redundant representations: a group with one grouped feature is used instead [4]. A detailed comparison of all these variants has been done by Schobbens et al. in [20]. The authors use a parameterized formal definition of the feature diagram, obtaining a framework useful for comparing and classifying all the variants, proving how the diverse options can be equivalent.

Most authors (see [7] for example) deal with the structural constraints implicit in the features tree (or graph) independently of the additional mutex/requires constraints. The definition of the complete feature models, therefore, requires working with graphs (the structure) and logical expressions (the constraints). Some recent works are devoted to the global validation of feature models, mainly based on propositional formulas [1] or constraint solvers [2].

Batory [1] uses a grammar and propositional formulas to validate the PL and each PL configuration. A sound connection between FDs, grammars, and propositional formulas was established and a system (*logic truth maintenance system*) enabling the propagation of constraints as user select features, to avoid inconsistent product specifications, was proposed. Furthermore, rules for transformation between feature models and grammars and a tool named GUIDSL, which takes a grammar as input and provides a graphical user interface to create configurations, was provided. Mendonça et al. use a two stage analysis to validate the models [17]. The advantage of using hypergraphs is the remarkable simplification of the supporting model. Instead of transforming FDs into a set of formulas to find inconsistencies or configure the final product, the algorithms can be used directly on the constrained hypergraphs, using a unique formal-

[4] http://giro.infor.uva.es/FeatureTool.html

ism. Modeling and transformation tools are consequently easier to define and implement.

6. Conclusions and Future Work

In this article, we have used F-hypergraphs to describe the abstract syntax of feature diagrams and their configuration. A valid feature diagram configuration is defined as a subset of its features, where the root is always present and the rest of the features satisfy two properties (at least one of its *structural* parents is present and for each leaving hyperarc at least minimum and at most maximum features of the hyperarc head are also present in the configuration). A configuration procedure has been defined and implemented.

Once the formal definition is stated, the construction of an extensible feature meta-model has been dealt with. The algebraic definition directly yields the required invariants, establishing a firm foundation for the meta-model. The advantages of simplicity and extensibility have made it possible to build modeling feature tools compatible with the different flavors of feature diagrams.

Work in progress includes the revised version of FMT, which will incorporate internally the proposed meta-model and the implemented configuration algorithms. The algorithms themselves are being optimized for their tool utilization, using FD preprocessing. Basically, the independent application of the first algorithm to each feature allows a sub-hypergraph (or an ordered set of them) to be associated with it, so that a configuration can be found faster by combining the sub-hypergraphs related to each feature (i.e., the union of features and hyperarc sets). If a valid partial configuration results, the second algorithm is applied to complete the configuration.

References

1. Batory, D.S.: Feature models, grammars, and propositional formulas. In: Obbink, J.H., Pohl, K. (eds.) Software Product Lines, 9th International Conference, SPLC 2005, Rennes, France, September 26-29, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3714, pp. 7–20. Springer (2005)
2. Benavides, D., Martín-Arroyo, P.T., Cortés, A.R.: Automated reasoning on feature models. In: Pastor, O., e Cunha, J.F. (eds.) Advanced Information Systems Engineering, 17th International Conference, CAiSE 2005, Porto, Portugal, June 13-17, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3520, pp. 491–503. Springer (2005)
3. Bosch, J.: Design and Use of Software Architectures: Adopting and Evolving a Product Line Approach. Pearson Education (Addison-Wesley & ACM Press) (May 2000)
4. Cechticky, V., Pasetti, A., Rohlik, O., Schaufelberger, W.: Xml-based feature modelling. In: Software Reuse: Methods, Techniques and Tools: 8th International Conference, ICSR 2004, Madrid, Spain, July 5-9, 2009. Proceedings. Lecture Notes in Computer Science, vol. 3107, pp. 101–114. Springer (2004)
5. Czarnecki, K., Eisenecker, U.W.: Generative Programming: Methods, Tools, and Applications. Addison-Wesley (2000)

6. Czarnecki, K., Helsen, S., Eisenecker, U.W.: Formalizing cardinality-based feature models and their specialization. Software Process: Improvement and Practice 10(1), 7–29 (2005)
7. Czarnecki, K., Helsen, S., Eisenecker, U.W.: Staged configuration through specialization and multilevel configuration of feature models. Software Process: Improvement and Practice 10(2), 143–169 (2005)
8. Gallo, G., Longo, G., Pallottino Sang, S.: Directed hypergraphs and applications. Discrete Applied Mathematics 42(2-3), 177–201 (1993)
9. Gallo, G., Scutella, M.: Directed hypergraphs as a modelling paradigm. Decisions in Economics and Finance 21(1), 97–123 (1998)
10. Griss, M.L., Favaro, J., d'Allessandro, M.: Integrating feature modeling with the RSEB. In: Devanbu, P., Poulin, J. (eds.) Proceedings: Fifth International Conference on Software Reuse. pp. 76–85. IEEE Computer Society Press (1998)
11. Kang, K., Cohen, S., Hess, J., Novak, W., Peterson, S.: Feature-oriented domain analysis (FODA) feasibility study. Tech. Rep. CMU/SEI-90-TR-21, Software Engineering Institute, Carnegie Mellon University (Nov 1990)
12. Kang, K.C., Kim, S., Lee, J., Kim, K., Shin, E., Huh, M.: Form: A feature-oriented reuse method with domain-specific reference architectures. Annals of Software Engineering 5, 143–168 (January 1998)
13. Laguna, M.A., González-Baixauli, B., Marqués Corral, J.M.: Seamless development of software product lines: Feature models to uml traceability. In: Sixth International Conference on Generative Programming and Component Engineering (GPCE 07). Salzburg, Austria. ACM Press (oct 2007)
14. Laguna, M.A., Marqués Corral, J.M.: Feature diagrams and their transformations: an extensible meta-model. In: IEEE proceedings of the EUROMICRO SEEA (2009)
15. Lau, S.: Domain analysis of e-commerce systems using feature-based model templates. Msc thesis, ECE Department, University of Waterloo, Canada (2006)
16. Mendonça, M., Branco, M., Cowan, D.D.: S.p.l.o.t.: software product lines online tools. In: Arora, S., Leavens, G.T. (eds.) Companion to the 24th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA 2009, October 25-29, 2009, Orlando, Florida, USA. pp. 761–762. ACM (2009)
17. Mendonça, M., Cowan, D.D., Malyk, W., de Oliveira, T.C.: Collaborative product configuration: Formalization and efficient algorithms for dependency analysis. Journal of Software 3(2), 69–82 (2008)
18. Riebisch, M., Böllert, K., Streitferdt, D., Philippow, I.: Extending feature diagrams with UML multiplicities. In: Proceedings of the Sixth Conference on Integrated Design and Process Technology (IDPT 2002). Pasadena, CA (jun 2002)
19. Rodríguez-Cano, G.: Configuration of Hyper-Graph based Feature Diagrams. Master's thesis, Department of Information Technology (2010), `http://uu.diva-portal.org`
20. Schobbens, P.Y., Heymans, P., Trigaux, J.C., Bontemps, Y.: Generic semantics of feature diagrams. Computer Networks 51(2), 456–479 (2007)

Miguel A. Laguna is Associate Professor in the Department of Computer Science at the University of Valladolid (Spain). He graduated and received a PhD from the University of Valladolid. He is the leader of the GIRO research group, specialized on systematic software reuse and software refactoring.

Jose Manuel Marques is Associate Professor in same Department. He graduated at the Complutense University of Madrid and obtained a PhD in Computer Science from the University of Valladolid. His research interests include OO modeling, software reuse and requirements engineering

Guillermo Rodrguez-Cano received his degree in Computer Science in 2008 from the University of Valladolid, and obtained a Master degree in 2010 at the Uppsala University, Sweden. His research interests include software product lines and model-driven architecture.

Received: August 4, 2010; Accepted: May 9, 2011

DOI: 10.2298/CSIS100906003W

Integrating Instance-level and Attribute-level Knowledge into Document Clustering

Jinlong Wang[1,3], Shunyao Wu[1], Gang Li[2], Zhe Wei[4,5]

[1] School of Computer Engineering, Qingdao Technological University
266033 Qingdao, China
{wangjinlong,shunyaowu}@gmail.com
[2] School of Information Technology, Deakin University
3125, Victoria, Australia
gang.li@deakin.edu.au
[3] Medical College of Qingdao University,
266021 Qingdao, China
[4] State Key Laboratory of CAD&CG, Zhejiang University
310027 Hangzhou, China
[5] SANYHE International Holding Co., Ltd,
110027 Shenyang, China

Abstract. In this paper, we present a document clustering framework incorporating instance-level knowledge in the form of pairwise constraints and attribute-level knowledge in the form of keyphrases. Firstly, we initialize weights based on metric learning with pairwise constraints, then simultaneously learn two kinds of knowledge by combining the distance-based and the constraint-based approaches, finally evaluate and select clustering result based on the degree of users' satisfaction. The experimental results demonstrate the effectiveness and potential of the proposed method.

Keywords: document clustering, pairwise constraints, keyphrases.

1. Introduction

Document clustering is one of the paramount tasks in text analysis and mining, for a wide range of information retrieval tasks, such as documents classification, documents summarization and visualization, *etc*. The traditional document clustering is unsupervised exploratory learning process, assuming no training samples from the user, automatically grouping unlabeled similar documents into meaningful clusters while separating documents with different topics. However, the performance of document clustering is usually unsatisfactory. There are many reasons, such as (1) the bag of words (BOW) model which is usually used in document clustering is relatively weak [11]; (2) it is unsupervised and impossible to interact with people; (3) it is difficult to understand the meaning of partitions sometimes.

In practice, there is usually some prior knowledge available for use, which can improve the clustering quality. Recently, many researchers have employed these prior knowledge to assist unsupervised document clustering, becoming a

hot topic in data mining and machine learning communities [3,4,5,14,7,11,12]. [4] proposed a probabilistic semi-supervised framework combining constraint-based and distance-based approaches with instance-level knowledge in the form of pairwise constraints. [3] proposed an effective method to actively obtain pairwise constraints based on [4] and [5]. [11] utilized Wikipedia as background knowledge to construct bag of concepts (BOC) model, and partitioned documents with pairwise constraints obtained by active learning. [12] proposed a semi-supervised clustering framework that actively selects informative pairwise constraints for obtaining user feedback.

Indeed, the semi-supervised document clustering approaches make use of additional information to increase clustering quality and make the partition easy to understand. Nevertheless, a majority of existing work are overwhelmed by attribute-level knowledge side information, except [2], which extracts keyphrases from *Title* and *Keywords*, and sets large weights to the keyphrases. Experimental results demonstrate the effectiveness of their method on short articles such as News. In addition, keyphrases can be obtained by utilizing some methods of keyphrase extraction or keyphrase assignment [22,13].

However, almost all the aforementioned approaches only incorporated one kind of knowledge. The performance of clustering quality with both kinds of side information becomes an interesting problem. In text classification, Vikas Sindhwani *etc*. proposed two classification algorithms that supported dual supervision in the form of labels for both examples and features in 2008 [16], and designed two strategies for active dual supervision in 2009 [15,17]. Experimental results demonstrate the effectiveness and potential of their algorithms.

In this paper, we aim to integrate both the instance-level knowledge in the form of pairwise constraints and the attribute-level knowledge in the form of keyphrases to assist document clustering. Based on the semi-supervised method integrating pair-wise constraints and attribute preferences [20], we present a framework for document clustering analysis. Firstly, we utilize pairwise constraints to construct optimization so as to obtain initial weights, then, we add keyphrases and simultaneously learn the two knowledge, finally, we evaluate and select the result according to the degree of users' satisfaction.

The rest of the article is organized as follows. In section 2, we introduce the two knowledge incorporated by our method, pairwise constraints and keyphrases; in section 3, we propose our framework incorporating pairwise constraints and keyphrases; we demonstrate experimental results in section 4; finally we conclude the paper in Section 5.

2. Notations

Given a set of n documents $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ with d words, where $\mathbf{x}_i = [x_{i1}, \cdots, x_{id}]^t$ (t denotes the transpose operation), $\mathbf{x}_i \in \mathbb{R}^d$, the desired number of clusters k, "must-link" set $\mathcal{S}$, "cannot-link" set $\mathcal{D}$ and keyphrases set $\mathcal{P}$, the objective of clustering is to obtain a partition of $\mathcal{X}$. In addition, $|\mathcal{S}|$ stands for the number of constraints in set $\mathcal{S}$.

2.1. Pairwise Constraints

Instance-level knowledge utilized by constrained clustering includes labels, pairwise constraints, *etc*. Considering the definition of traditional clustering and our strategy to incorporate dual knowledge, this paper chooses pairwise constraints as instance-level knowledge.

The set of pairwise constraints comprises "must-link" set $\mathcal{S}$ and "cannot-link" set $\mathcal{D}$.

- $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}$ means $\mathbf{x}_i$ and $\mathbf{x}_j$ are in the same cluster.
- $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{D}$ means $\mathbf{x}_i$ and $\mathbf{x}_j$ are in different clusters.

2.2. Keyphrases

Keyphrases provide brief summaries of documents' content and reflect main topic of documents [22], such as words in title, keywords, MeSH (Medical Subject Headings) information in biomedical texts, *etc*. There are many different types of approaches to obtain keyphrases, such as keyphrase extraction, keyphrase assignment, and so on. In this paper, we extract keyphrases from *Title* and *Keywords* [2], and utilize attribute order preferences [18] to express keyphrases.

An attribute order preference (s, t, δ) ($\delta > 0$) stands for $\mathbf{w}_s - \mathbf{w}_t \geq \delta$. This means that the attribute s is more important than the attribute t. However, it is complicated to exactly specify how much term s is important than term t in document clustering. Thus, we define keyphrases as (s, δ) ($\mathbf{w}_s \geq \delta$) and set a large enough value for δ.

2.3. Bregman divergences

For the consideration of expansibility, we incorporate Bregman divergences into our framework. The Bregman divergences [1] include many useful distance metrics, such as squared Euclidean distance, Mahalanobis distance, KL divergence, generalized I-divergence, *etc*.

Definition 1. *Provided that $\phi : \mathcal{S} \to \mathbb{R}$ is a strictly convex function defined on a convex set $\mathcal{S} \subseteq \mathbb{R}^d$ so that ϕ is differentiable on $ri(\mathcal{S})$ (the relative interior of $\mathcal{S}$). The Bregman divergences d_ϕ is defined as*

$$d_\phi(\mathbf{x_i}, \mathbf{x}_j) = \phi(\mathbf{x}_i) - \phi(\mathbf{x}_j) - < \mathbf{x}_i - \mathbf{x}_j, \nabla\phi(\mathbf{x}_j) >$$

where $\nabla\phi$ is the gradient vector of ϕ.

We can obtain different divergences by setting a different function ϕ. Given $\phi(\mathbf{x}) = \mathbf{x}^T A\mathbf{x}$, we can have $d_\phi(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)A(\mathbf{x}_i - \mathbf{x}_j)$; or when given $\phi(\mathbf{x}) = \sum_{m=1}^{d} x_m log x_m$, we have $d_\phi(\mathbf{x}_i, \mathbf{x}_j) = \sum_{m=1}^{d} x_{im} log \frac{x_{im}}{x_{jm}} - \sum_{m=1}^{d}(x_{im} - x_{jm})$.

There are many types of distances for document clustering, such as cosine similarity, KL divergence, generalized I-divergence, *etc*. In order to facilitate solving optimization problem constructed based on metric learning, this

paper considers to use generalized I-divergence as the distance metric. Since I-divergence is not symmetric, we will modify it to "I-divergence to the mean", d_{IM} [4].

$$d_{IM}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{m=1}^{d} x_{im} log \frac{2x_{im}}{x_{im} + x_{jm}} + \sum_{m=1}^{d} x_{jm} log \frac{2x_{jm}}{x_{im} + x_{jm}}$$

Then, we parameterize the above distance by a vector of non-negative weights $\mathbf{w}$:

$$d_{IM_{\mathbf{w}}}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{m=1}^{d} w_m x_{im} log \frac{2x_{im}}{x_{im} + x_{jm}} + \sum_{m=1}^{d} w_m x_{jm} log \frac{2x_{jm}}{x_{im} + x_{jm}}$$

3. A Semi-supervised Document Clustering Framework

In this section, we will propose the document clustering framework which incorporates pairwise constraints and keyphrases. Given a document repository and the two kinds of prior knowledge, our approach deals with the problem of effectively incorporating them with the appropriate distance learning. In general, the steps of our approach are as follows:

1. Incorporate pairwise constraints to initialize weights based on metric learning.
2. Add keyphrases to simultaneously learn the two knowledge combining constrained-based and distance-based approaches.
3. Evaluate and select clustering result according to the degree of users' satisfaction.

3.1. Initialize Weights Based on Metric Learning with Pairwise Constraints

Obtaining good initial weights is important to metric leaning, thus we initialize weights according to Halkidi's approach [8]. We construct optimization with pairwise constraints according to Xing's thought [24] so as to make sure must-link pair documents as similar as possible.

$$\min_{\mathbf{w}} \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}} D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{x}_j) - \lambda H(\mathbf{w})$$

$$\textit{subject to:} \quad \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{D}} D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{x}_j) \geq 1$$

$$\mathbf{w} \in \mathbb{R}_{+}^{d} \tag{1}$$

3.2. Learn the Two Knowledge Combining Constraint-based and Distance-based Approaches Simultaneously

Through solving the optimization problem (1), we can obtain initial weights $\mathbf{w}_{initial}$. After that, we aim to simultaneously learn the two knowledge and the objective function is as follows:

$$min_{\mathbf{w},\mu,\pi} \frac{1}{n} \sum_{c=1}^{k} \sum_{x_i \in \pi_c} D_{\mathbf{w}}(\mathbf{x}_i, \mu_c) + \lambda_1 \sum_{(s,\delta)\in\mathcal{P}} max(\delta - w_s, 0) +$$

$$\lambda_2 \Phi_{pairwise\_constraints} - \lambda_3 H(\mathbf{w})$$

The first term is an objective clustering validation index, intra-cluster distortion of the clusters $\{\pi_c\}_{c=1}^{k}$; the second term is the penalty term of keyphrases which represents the satisfactory of attribute weights for keyphrases; the third term stands for the penalty term of pairwise constraints; the last term is the regularization term which guarantees the consistence of attribute weights.

The third term includes the penalty of must-link constraints and cannot-link constraints. According to [4], we set $(\sum \phi(\mathbf{x}_i \neq \mathbf{x}_j) D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{x}_j)/|\mathcal{S}_{unsat}|)^2$ for the penalty of must-link constraints and $(\sum \phi(\mathbf{x}_i = \mathbf{x}_j)(D_{\mathbf{w}_{max}} - D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{x}_j)/|\mathcal{D}_{unsat}|)^2$ for cannot-link constraints.

$$\Phi_{pairwise\_constraints} = (\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} \phi(\mathbf{x}_i \neq \mathbf{x}_j) D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{x}_j)/|\mathcal{S}_{unsat}|)^2 +$$

$$(\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}} \phi(\mathbf{x}_i = \mathbf{x}_j)(D_{\mathbf{w}_{max}} - D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{x}_j))/|\mathcal{D}_{unsat}|)^2$$

Here, $\phi(True) = 1$ and $\phi(False) = 0$; $\mathbf{x}_i \neq \mathbf{x}_j$ stands for cluster index of $\mathbf{x}_i$ unequal to $\mathbf{x}_j$ ($\mathbf{x}_i \in \pi_c$ and $\mathbf{x}_j \notin \pi_c$), while $\mathbf{x}_i = \mathbf{x}_j$ stands for cluster index of $\mathbf{x}_i$ equal to $\mathbf{x}_j$ ($\mathbf{x}_i \in \pi_c$ and $\mathbf{x}_j \in \pi_c$); $D_{\mathbf{w}_{max}}$ stands for the maximum distance between two arbitrary points for the dataset; $|\mathcal{S}_{unsat}|$ stands for number of unsatisfied must-link constraints while $|\mathcal{D}_{unsat}|$ stands for number of unsatisfied cannot-link constraints. The higher the satisfaction level, the lower the penalty term.

In order to ensure that attribute weights are uniform, we use l_2 *entropy* as the regularization term and set $H(\mathbf{w}) = 1 - \mathbf{w}^T\mathbf{w}$.

There are three variables in the optimization problem, and it is impossible to solve it directly. Thus, we use EM framework to deal with the problem and design three steps. Firstly, given $\{\mu_c\}_{c=1}^{k}$ and $\mathbf{w}$, assign each data point to minimize objective function; then, given $\{\pi_c\}_{c=1}^{k}$, re-calculate cluster centroids $\{\mu_c\}_{c=1}^{k}$; finally, given $\{\pi_c\}_{c=1}^{k}$ and $\{\mu_c\}_{c=1}^{k}$, solve the optimization problem to obtain $\mathbf{w}$. Iterate until convergence.

E-step In simple *k*-means clustering, the E-step assigns each point to the nearest cluster centroid given a certain clustering distance metric. There are

Algorithm 1: The procedure of clustering with the two knowledge

Data: Dataset $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$, number of output clusters k, initial weights $\mathbf{w_{initial}}$, must-link constraints $\mathcal{S}$, cannot-link constraints $\mathcal{D}$, and keyphrases $\mathcal{P}$.

Result: Clusters obtained with pairwise constraints and keyphrases.

begin

1. Initialize k cluster representatives $\{\mu_c\}_{c=1}^k$ and set $\mathbf{w} = \mathbf{w_{initial}}$;

2. repeat

2-a. **E-step**: Given $\{\mu_c\}_{c=1}^k$ and $\mathbf{w}$, re-assign data points to clusters to obtain $\{\pi_c\}_{c=1}^k$.

2-b. **M-step(A)**: Given $\{\pi_c\}_{c=1}^k$, re-calculate cluster centroids $\{\mu_c\}_{c=1}^k$.

2-c. **M-step(B)**: Given $\{\pi_c\}_{c=1}^k$ and $\{\mu_c\}_{c=1}^k$, re-estimate $\mathbf{w}$ by solving the optimization problem.

until convergence

3. return $\{\pi_c\}_{c=1}^k$.

also some other methods, such as iterated conditional models (ICM) in [4] that treated objective function as optimization problem to solve, evolutionary algorithm [9] and so on.

When given $\{\mu_c\}_{c=1}^k$ and $\mathbf{w}$, the objective function is transformed into:

$$J_\pi = min_\pi \frac{1}{n} \sum_{c=1}^{k} \sum_{x_i \in \pi_c} D_{\mathbf{w}}(\mathbf{x}_i, \mu_c) + \lambda_2 \Phi_{pairwise\_constraints}$$

Thus, this paper solves the optimization problem by ICM approach to obtain cluster assignments. Firstly, the ICM algorithm sets random order for each point; then, assign each point to the cluster centroid which minimizes the above objective function J_π. Iterate until convergence ($\{\pi_c\}_{c=1}^k$ does not change or J_π dose not obviously decrease between two sequential iterations).

M-step(A) The M-step(A) is one step of the M-step to re-estimate cluster centroids $\{\mu_c\}_{c=1}^k$. [1] has shown each cluster centroid re-estimated in M-step is the arithmetic mean of the points in that cluster. Thus, we calculated cluster centroids in *k*-means clustering with squared Euclidian distance as the formula: $\mu_k^{squared} = \frac{\sum_{\mathbf{x}_i \in \pi_c} \mathbf{x}_i}{|\pi_c|}$. Different from the squared Euclidian distance, given I-divergence, cluster centroids are re-estimated as follows:

$$\mu_k^{IM} = \alpha \frac{\sum_{\mathbf{x}_i \in \pi_c} \mathbf{x}_i}{|\pi_c|} + (1 - \alpha) \frac{1}{n}$$

Here, α ($0 < \alpha < 1$, such as $\alpha = 0.9$) is a smoothing factor to guarantee the denominator of $log \frac{2x_{im}}{x_{im} + \mu_{km}^{IM}}$ in $d_{I_{\mathbf{w}}}(\mathbf{x}_i, \mu_k^{IM})$ is unequal to 0.

M-step(B) The M-step(B) aims to compute weight by solving optimization constructed according to the objective function provided that $\{\pi_c\}_{c=1}^k$ and $\{\mu_c\}_{c=1}^k$ are given.

$$min_{\mathbf{w}} \frac{1}{n} \sum_{c=1}^{k} \sum_{x_i \in \pi_c} D_{\mathbf{w}}(\mathbf{x}_i, \mu_c) + \lambda_1 \sum_{(s,\delta)\in\mathcal{P}} max(\delta - w_s, 0) + \lambda_2 \Phi_{pairwise\_constraints} + \lambda_3 \mathbf{w}^T \mathbf{w}$$

$$\begin{aligned} \textit{subject to:} \quad & for\ each\ important\ word\ p \in P, w_s \geq \delta \\ & w_1 + ... + w_d = 1 \\ & \sum_{(\mathbf{x}_i, \mathbf{x}_j)\in\mathcal{D}} D_{\mathbf{w}}(\mathbf{x}_i, \mathbf{x}_j) \geq 1 \\ & \mathbf{w} \in \mathbb{R}_+^d \end{aligned} \tag{2}$$

The problem (formula 2) is a convex optimization according to [4,18], and there are many effective algorithms to solve the optimization, such as newton method, homogeneous algorithm, active set method an so on [6]. We utilize MOSEK package [6] to solve optimization problems (formula 1 and formula 2).

3.3. Evaluate and Select Clustering Result Based on the Degree of Users' Satisfaction

Degree of users' satisfaction is the portion of knowledge that is satisfied in the clustering result. Many researchers utilize the degree of users' satisfaction to evaluate intermediate results and further improve clustering quality [8,21]. Generally, we think large degree of users' satisfaction can reflect good clustering quality. Thus, in this paper, we wish our approach can effectively incorporate the two knowledge so that the degree of users' satisfaction is satisfied.

$$\begin{aligned} accuracy &= accuracy_{pairwise\ constraints} + accuracy_{important\ words} \\ &= (|sat(\mathcal{S})| + |sat(\mathcal{D})|)/(|\mathcal{S}| + |\mathcal{D}|) + |sat(\mathcal{P})|/|\mathcal{P}| \end{aligned}$$

Here, $sat(*)$ means the satisfied constraints in the set *.

In this paper, the degree of users' satisfaction includes satisfaction of pairwise constraints and keyphrases. We utilize Sun's approach [18] to set parameters in objective function, and all the keyphrases information can be satisfied. Thus, we should lay stress on pairwise constraints. As we integrate two knowledge, our approach should be better than those only incorporating pairwise constraints, and our approach on satisfaction of pairwise constraints should be also better. Even if worse, it should not be much lower.

However, there are many complicated issues when incorporating these two kinds of knowledge, such as the conflicting information, suitable initial cluster centroids [21], *etc*. Thus, our approach is not always optimal on satisfaction of

[6] http://www.mosek.com/

pairwise constraints. As a heuristic, when decrease performance is observed (5% decrease is observed in this paper), we think it is inappropriate to incorporate the two knowledge, and utilize clustering result of metric learning method only with pairwise constraints in section 3.1 as final result.

3.4. Time Complexity

Let N be the number of documents in the collection. The first step includes two parts, constructing and solving the optimization problem with pairwise constraints to obtain new metric and partition documents by utilizing new metric. Time complexity of constructing and solving the optimization problem is related with number of pairwise constraints. Hence, its complexity is estimated to be $O(N)$ [8]. Then, given a clustering algorithm Alg, such as EM hard clustering algorithm utilized in this paper, we can partition the documents by new metric.

The main work of the second step is utilizing a variant EM clustering algorithm to partition documents. Different from the unsupervised version, we utilize ICM approach to assign each document in E-Step, and add M-Step(B) to optimize the distance, solving the optimization problem with two types of knowledge. Let $Complexity(ICM)$ be time complexity of ICM approach, and t be iterations of EM clustering algorithm. Thus, the time cost of the second step is estimated to be $O(Complexity(Alg) + t * (Complexity(ICM) + N))$.

The third step is just a simple comparison, and we can ignore its complexity.

According to the preceding analysis, the complexity of our approach is $O(Com plexity(Alg) + t * Complexity(ICM) + t * N)$. Usually, $t << N$ and the complexity of ICM approach is very low. Hence, the time complexity of our approach mainly depends on the complexity of the clustering algorithm.

4. Experimental Results

In this section, we demonstrate experimental results of our approach comparing with *k*-means, Xing's method [24] and CFP algorithm [18] on *20Newsgroups* collection.

1. *k*-means algorithm is unsupervised and only depends on objective criteria to partitions documents.
2. Xing's method [24] constructs optimization to learn pairwise constraints, and utilizes obtained new metric to partition documents. In this paper, we solve the optimization problem in section 3.1 to obtain new metric.
3. CFP algorithm [18] incorporates keyphrases (attribute order preferences) to assist document clustering. In this paper, we utilize EM framework in section 3.2 to integrate keyphrases while the objective function is as follows:

$$min_{\mathbf{w},\mu,\pi} \frac{1}{n} \sum_{c=1}^{k} \sum_{x_i \in \pi_c} D_{\mathbf{w}}(\mathbf{x}_i, \mu_c) + \lambda_1 \sum_{(s,\delta)\in\mathcal{P}} max(\delta - w_s, 0) - \lambda_3 H(\mathbf{w})$$

4. Our method integrates pairwise constraints and keyphrases into document clustering.

4.1. Datasets and Experimental Settings

We derive 3 datasets from *20Newsgroups* collection. We randomly select 100 documents for each category from original dataset, and derive 3 datasets with 3 categories, News_Different_3 (alt.atheism, rec.sport.baseball, and sci.space) including 3 newsgroup on different topics, News_Related_3 (talk.politics.misc, talk.politics.guns, and talk.politics.mideast) with relevant topics and News_Similar_3 (comp.graphics, comp.os.ms-windows, and comp.windows.x) with large overlap among each category.

We remove stop words, high-frequency and low-frequency words, and express each dataset by TFIDF weighting. Finally, we normalize each dataset so as to avoid impact of document length and make the dissimilarity among documents clearer [23]. Each text vector, $< tf_1 log(\frac{|D|}{df_1}), ..., tf_d log(\frac{|D|}{df_d}) >$, is normalized as follows:

$$< \frac{tf_1 log(\frac{|D|}{df_1})}{\sqrt{(tf_1 log(\frac{|D|}{df_1}))^2 + ... + (tf_d log(\frac{|D|}{df_d}))^2}}, ..., \frac{tf_d log(\frac{|D|}{df_d})}{\sqrt{(tf_1 log(\frac{|D|}{df_1}))^2 + ... + (tf_d log(\frac{|D|}{df_d}))^2}} >$$

As the keyphrases in each derived dataset are few and we want to provide enough keyphrases to assist document clustering, we treat the whole *20Newsgroups* collection as background knowledge, and extract keyphrases from categories that each derived dataset belongs to. In this way, we can obtain many keyphrases, and further select some keyphrases with high word frequency (we select $\lfloor \frac{d}{4} \rfloor$ keyphrases in experiments).

For reliability of experimental results, we make 2-fold cross-validation for each dataset [19,4,5]. We randomly select pairwise constraints from 50% of the dataset, and test methods on remaining 50%. For robustness of experimental results, clustering accuracy is averaged using 10 runs with randomly selected pairwise constraints.

In addition, we set $\lambda_1 = \frac{d}{|\mathcal{P}|}$, $\lambda_2 = 1$ and $\lambda = \lambda_3 = d$ so as to make sure three terms to contribute equally to the objective value [18], $\delta = 4/d$, and $d_{IM} = [\underbrace{1, .., 1}_{d}]$ for $D_{\mathbf{w}_{max}}$ (After normalizing dataset, the value range of attributes becomes [0,1]).

4.2. Evaluation Criteria

In this paper, we utilize two common indexes in document clustering, Purity and Normalized Mutual Information (NMI) to evaluate clustering quality.

Purity measures how close the cluster assignment versus underlying class labels by building one to one correspondence between the clusters and the classes.

$$Purity(\mathcal{C}, \mathcal{B}) = \frac{max_{Map(i) \in [1,..,k]} \sum_{i=1}^{k} n_{i,Map(i)}}{n}$$

Here, $\mathcal{C}$ stands for random variables denoting the clustering assignment while $\mathcal{B}$ presents random variables for the pre-specified class labels. The number of groups in $\mathcal{C}$ and $\mathcal{B}$ are both k. n stands for number of documents in the corpus, and i stands for the cluster index. $Map(i)$ is the class label corresponding to the cluster index i, and $n_{i,Map(i)}$ is the number of documents not only belonging to cluster i but class $Map(i)$.

Normalized Mutual Information (NMI) is an effective index based on information theory.

$$NMI(\mathcal{C},\mathcal{B}) = \frac{I(\mathcal{C};\mathcal{B})}{\sqrt{H(\mathcal{C})H(\mathcal{B})}} = \frac{\sum_{i=1}^{k}\sum_{j=1}^{k} n_{ij} log\frac{n\cdot n_{ij}}{n_i\cdot n'_j}}{\sqrt{\sum_{i=1}^{k} n_i log\frac{n_i}{n}\sum_{j=1}^{k} n'_j log\frac{n'_j}{n}}}$$

Here, n_i presents the document number in the ith cluster of $\mathcal{C}$, n'_j denotes the document number in the jth class of $\mathcal{B}$. n_{ij} denotes the item number included in ith cluster and jth class.

4.3. Results Comparison

Comparison to Other Methods Table 1 and Table 2 demonstrate the result comparisons under the Purity and NMI indexes. Overall, our approach is obviously better than other methods. Especially on the News_Similar_3 dataset, our approach increases 10% under Purity index with a small amount of prior knowledge.

Table 1. Our approach versus Competing methods under Purity index with 30 pairwise constraints (15 must-link constraints and 15 cannot-link constraints) and $\lfloor\frac{d}{4}\rfloor$ keyphrases

Datasets	*k*-means	Xing's method	CFP algorithm	Our method
News_Different_3	0.8160 ± 0.0872	0.9027 ± 0.0412	0.9260 ± 0.0438	0.9400 ± 0.0231
News_Related_3	0.6427 ± 0.0530	0.6800 ± 0.0514	0.6847 ± 0.0655	0.7467 ± 0.0916
News_Similar_3	0.4547 ± 0.0584	0.4747 ± 0.0603	0.5313 ± 0.0595	0.5847 ± 0.0784

Clustering Accuracy versus Constraints We keep the number of keyphrases as $\lfloor\frac{d}{4}\rfloor$ and get results with number of pairwise constraints increasing. The number of pairwise constraints in Fig. 1, 2 and 3, m stands for m must-link constraints and m cannot-link constraints.

Table 2. Our approach versus Competing methods under NMI index with 30 pairwise constraints (15 must-link constraints and 15 cannot-link constraints) and $\lfloor \frac{d}{4} \rfloor$ keyphrases

Datasets	*k*-means	Xing's method	CFP algorithm	Our method
News_Different_3	0.5591 ± 0.0965	0.6919 ± 0.1007	0.7680 ± 0.0765	0.7858 ± 0.0660
News_Related_3	0.3307 ± 0.0850	0.3651 ± 0.0662	0.4311 ± 0.0758	0.4923 ± 0.0829
News_Similar_3	0.0568 ± 0.0410	0.0859 ± 0.0621	0.1188 ± 0.0557	0.1779 ± 0.0744

k-means and CFP algorithm do not incorporate pairwise constraints, their clustering quality should not be affected by pairwise constraints. However, their performances are all unstable. It is mainly due to the initialization of cluster centroids. Even so, as shown in Fig. 1, 2 and 3, CFP algorithm is always much better than *k*-means. It illuminates that incorporating keyphrases extracted from *Title* and *Keywords* can increase document clustering quality.

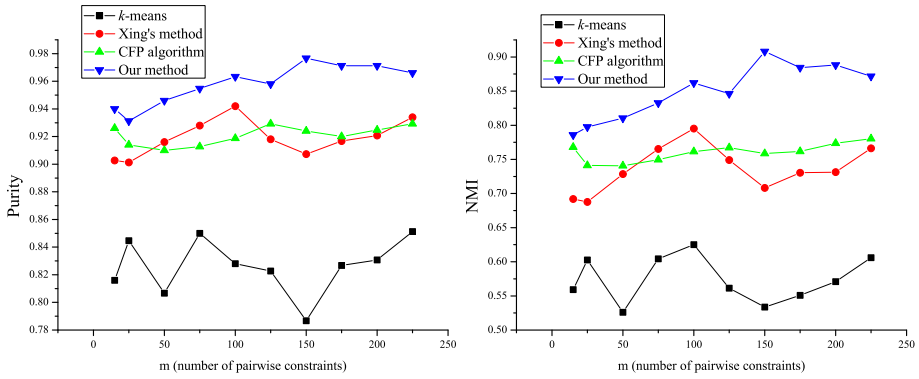


Fig. 1. Clustering accuracy on News_Different_3 with number of pairwise constraints increasing

In Fig. 1, the performance of our approach is obviously better than Xing's method, and even can improves about 10% under NMI index when number of pairwise constraints is few. It is mainly because topics in News_Different_3 dataset are easy to distinguish and keyphrases can effectively reflect topics. The topic of alt.atheism is religion, atheism, *etc.*, rec.sport.baseball is basketball, and sci.space is astrospace, universal gravitation, *etc.* As shown in Table 3, keyphrases of News_Different_3 can be directly matched with corresponding topics. For example, "atheists", "morality", "islamic", "christian", *etc.* should belong to alt.atheism, while "sky" and "moon" belong to sci.space. With the effec-

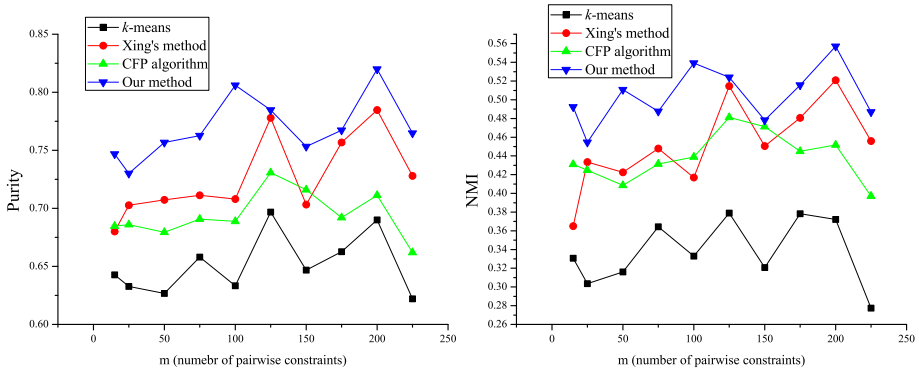


Fig. 2. Clustering accuracy on News_Related_3 with number of pairwise constraints increasing

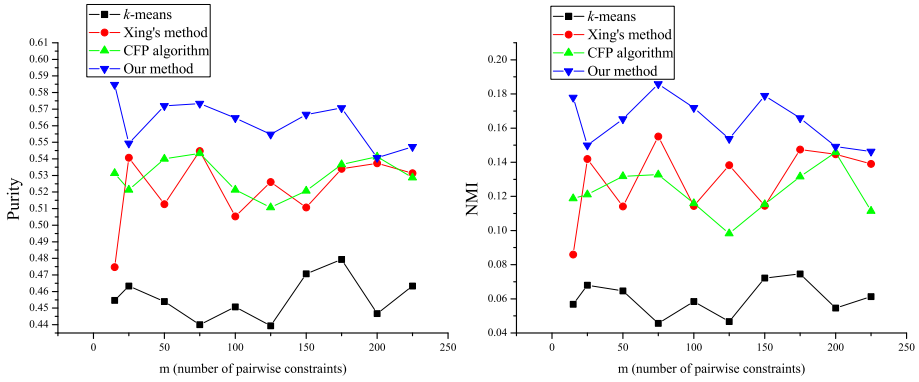


Fig. 3. Clustering accuracy on News_Similar_3 with number of pairwise constraints increasing

tive keyphrases, the performance of CFP algorithm is better than Xing's method under NMI index in most cases.

In Fig. 2 and Fig. 3, our method is slightly better than Xing's method. On the one hand, it is due to correlation and confused topics of the two datasets. There are many related and overlap among three categorization of News_Related_3 (talk.politics.misc, talk.politics.guns, and talk.politics.mideast), such as topic "gun" may appear in each categorization. Similar with News_Related_3, the documents of News_Similar_3 is mainly about computer help problems, and it is hardly to distinguish. On the other hand, keyphrases extracted from *Title* and *Keywords* is also hard to be assigned to corresponding topics. For example, as show in Table 3, keyphrases of News_Similar_3, "help", "do", "file", "problem", *etc*. belong to all the three categorization topics.

Table 3. Top 10 keyphrases (sorted by word frequency) of three datasets

News_Different_3	News_Related_3	News_Similar_3
atheists	waco	help
political	gun	do
morality	atf	dos
islamic	Clinton	window
baseball	burns	microsoft
update	ranch	file
sky	israel	win
moon	survivors	problem
players	gay	ms
christian	israeli	need

Table 4 and Table 5 show the *t*-Test [10] of our approach versus competing methods under Purity and NMI. When the probability is lower than 5%, it demonstrates the robustness of our approach is good and the performance of our method is obviously better than other method; However, when the probability is larger than 5%, it illuminates our approach is similar with other method. As shown in Table 4 and Table 5, our approach is obviously better than *k*-means, Xing's method and CFP algorithm on three datasets.

Table 4. *t*-Test: Our method versus Competing methods under Purity index

	k-means	Xing's method	CFP algorithm
News_Different_3	1.0723e-007	1.5592e-005	1.0886e-005
News_Related_3	1.4329e-007	6.4472e-004	1.1223e-005
News_Similar_3	3.4242e-008	0.0038	1.4409e-004
total	3.7362e-021	7.9132e-010	1.6270e-010

Table 5. *t*-Test: Our method versus Competing methods under NMI index

	k-means	Xing's method	CFP algorithm
News_Different_3	7.8849e-008	1.7325e-005	3.3633e-005
News_Related_3	2.7749e-008	0.0031	1.4376e-004
News_Similar_3	3.3062e-008	0.0062	8.1523e-005
total	4.6540e-013	6.2477e-008	5.5935e-011

Time Complexity Evaluation Fig. 4 shows the time complexity of our approach with respect to the size of collection and the number of pairwise constraints. In Fig. 4(a), we present results using a 3000-dimensional dataset with 100 randomly selected pairwise constraints (50 must-link constraints and 50 cannot-link constraints) and $\lfloor \frac{d}{4} \rfloor$ keyphrases. As shown in Fig. 4(a), the time complexity of our approach is nearly linear to the number of documents in the dataset. In addition, Fig. 4(b) shows the time cost increases linearly with number of pairwise constraints.

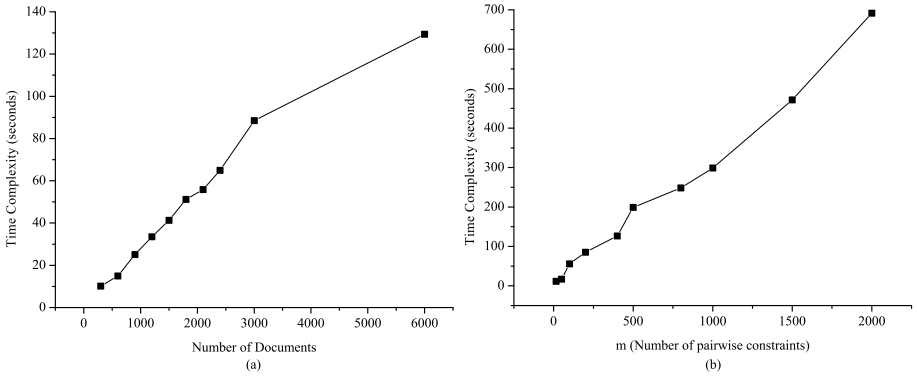


Fig. 4. Time complexity of our approach versus: (a) number of documents; (b) number of pairwise constraints.

5. Conclusion

This paper presents an effective semi-supervised document clustering framework for incorporating pairwise constraints and keyphrases. Our framework initializes attribute weights based on metric learning with pairwise constraints firstly, then simultaneously learn the two knowledge, finally evaluate and select clustering result according to the degree of users' satisfaction. The experimental results validate our method.

Our method can effectively integrate pairwise constraints and keyphrases into document clustering. It not only meets users' need but improve clustering quality. Even with few knowledge, the performance of our method is still satisfied. Moreover, document clustering with keyphrases should be paid much attention to, and its performance is better than clustering with pairwise constraints when keyphrases can effectively reflect document topics.

However, there are many parts to be improved. For simplicity, we set the same value ($\delta = \frac{4}{d}$) for all keyphrases, and it should treat keyphrases according to some criterions, such as word frequency. Secondly, how to solve the contradiction between keyphrases and pairwise constraints should be taken into

account. In addition, we should select suitable center centroids for CFP algorithm and our method, so as to improve the accuracy and robustness.

Acknowledgments. This work was partially supported by the National Natural Science Foundation of P.R.China (No.60802066, No.51005202, No.61004104), the China Postdoctoral Science Foundation (No.20100471494), the Excellent Young Scientist Foundation of Shandong Province of China under Grant (No.2008BS01009), and Deakin CRGS Grant 2011.

References

1. Banerjee, A., Merugu, S., Dhillon, I., Ghosh, J.: Clustering with bregman divergences. The Journal of Machine Learning Research 6, 1705–1749 (2005)
2. Banerjee, S., Ramanathan, K., Gupta, A.: Clustering short texts using wikipedia. In: SIGIR '07: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 787–788 (2007)
3. Basu, S., Banjeree, A., Mooney, E., Banerjee, A., Mooney, R.J.: Active semi-supervision for pairwise constrained clustering. In: SDM' 04: Proceedings of the 4th SIAM International Conference on Data Mining. pp. 333–344 (2004)
4. Basu, S., Bilenko, M., Mooney, R.J.: A probabilistic framework for semi-supervised clustering. In: KDD '04: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 59–68 (2004)
5. Bilenko, M., Basu, S., Mooney, R.J.: Integrating constraints and metric learning in semi-supervised clustering. In: ICML '04: Proceedings of the 21st International Conference on Machine learning. p. 11 (2004)
6. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge Univversity Press (2004)
7. Chen, Y., Rege, M., Dong, M., Hua, J.: Incorporating user provided constraints into document clustering. In: ICDM '07: Proceedings of the 7th IEEE International Conference on Data Mining. pp. 103–112 (2007)
8. Halkidi, M., Gunopulos, D., Vazirgiannis, M., Kumar, N., Domeniconi, C.: A clustering framework based on subjective and objective validity criteria. ACM Transactions on Knowledge Discovery from Data (TKDD) 1(4), 4 (2008)
9. Handl, J., Knowles, J.: An evolutionary approach to multiobjective clustering. IEEE Transactions on Evolutionary Computation 11(1), 56–76 (2007)
10. Hogg, R., Craig, A., McKean, J.: Introduction to mathematical statistics. Macmillan New York (1959)
11. Huang, A., Milne, D., Frank, E., Witten, I.H.: Clustering documents with active learning using wikipedia. In: ICDM '08: Proceedings of the 8th IEEE International Conference on Data Mining. pp. 839–844 (2008)
12. Huang, R., Lam, W.: An active learning framework for semi-supervised document clustering with language modeling. Data and Knowledge Engineering 68(1), 49–67 (2009)
13. Hulth, A., Karlgren, J., Jonsson, A., Bostroem, H., Asker, L.: Automatic keyword extraction using domain knowledge. In: CICLing '01 Proceedings of the 2nd International Conference on Computational Linguistics and Intelligent Text Processing. pp. 472–482 (2001)

14. Ji, X., Xu, W.: Document clustering with prior knowledge. In: SIGIR '06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 405–412 (2006)
15. Melville, P., Sindhwani, V.: Active dual supervision: reducing the cost of annotating examples and features. In: HLT '09: Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing. pp. 49–57 (2009)
16. Sindhwani, V., Hu, J., Mojsilovic, A.: Regularized co-clustering with dual supervision. In: NIPS '08: Neural Information Processing Systems. vol. 21 (2008)
17. Sindhwani, V., Melville, P., Lawrence, R.D.: Uncertainty sampling and transductive experimental design for active dual supervision. In: ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning. pp. 953–960 (2009)
18. Sun, J., Zhao, W., Xue, J., Shen, Z., Shen, Y.: Clustering with feature order preferences. In: PRICAI '08: Proceedings of the 10th Pacific Rim International Conference on Artificial Intelligence. pp. 382–393 (2008)
19. Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S.: Constrained k-means clustering with background knowledge. In: ICML '01: Proceedings of the 18th International Conference on Machine Learning. pp. 577–584 (2001)
20. Wang, J., Wu, S., Li, G.: An effective semi-supervised clustering framework integrating pair-wise constraints and attribute preferences. Computing and Informatics (in press)
21. Wang, J., Wu, S., Vu, H.Q., Li, G.: Text document clustering with metric learning. In: SIGIR '10: Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 783–784 (2010)
22. Witten, I., Paynter, G., Frank, E., Gutwin, C., Nevill-Manning, C.: KEA: Practical automatic keyphrase extraction. In: JCDL '99: Proceedings of the 4th ACM Conference on Digital Libraries. p. 255 (1999)
23. Wu, S., Wang, J., Vu, H.Q., Li, G.: Text clustering with important words using normalization. In: JCDL '10: Proceedings of the 10th Annual Joint Conference on Digital Libraries. pp. 393–394 (2010)
24. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.J.: Distance metric learning with application to clustering with side-information. In: NIPS '02: Neural Information Processing Systems. pp. 505–512 (2002)

Wang, Jinlong is an associate professor at School of Computer Engineering, Qingdao Technological University. He received the Diploma and Ph.D. degree at College of Computer Science and Technology from Zhejiang University, China in 2002 and 2007 respectively. His research interests include data mining, machine learning and artificial intelligence.

Wu, Shunyao is currently a master student at School of Computer Engineering, Qingdao Technological University.

Li, Gang is a lecturer in the School of Information Technology, Deakin University. He completed a PhD in 2005 at Deakin University in the area of data mining, and received the bachelor's degree in computer science from Xi'an Petroleum Institute in 1994, the master by research degree from Shanghai University of Science and technology in 1997. His research interests include data mining, wireless sensor networks and sentiment mining from multimedia.

Wei, Zhe is the ClO of SANYI Heavy Equipment CO. LTD. He received the Ph.D. degree at College of Mechanical Engineering from Zhejiang University, China in 2009. His research interests include mechanical design, product data management and so on.

Received: September 6, 2010; Accepted: February 14, 2011

DOI: 10.2298/CSIS100327022M

Code Cache Management Based on Working Set in Dynamic Binary Translator

Ruhui Ma, Haibing Guan*, Erzhou Zhu, Yongqiang Gao, and Alei Liang

Shanghai key laboratory of scalable computing and systems,
Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai, 200240,China,
{ruhuima, hbguan, ezzhu, yongqianggao, liangalei}@sjtu.edu.cn

Abstract. Software code cache employed to store translated or optimized codes, amortizes the overhead of dynamic binary translation via reusing of stored-altered copies of original program instructions. Though many conventional code cache managements, such as Flush, Least-Recently Used (LRU), have been applied on some classic dynamic binary translators, actually they are so unsophisticated yet unadaptable that it not only brings additional unnecessary overhead, but also wastes much cache space, since there exist several noticeable features in software code cache, unlike pages in memory. Consequently, this paper presents two novel alternative cache schemes—SCC (Static Code Cache) and DCC (Dynamic Code Cache) based on working set. In these new schemes, we utilize translation rate to judge working set. To evaluate these new replacement policies, we implement them on dynamic binary translator—CrossBit with several commonplace code cache managements. Through the experiment results based on benchmark SPECint 2000, we achieve better performance improvement and cache space utilization ratio.

Keywords: Code cache management, Working set, Replacement strategy, Code block, Bounded code cache.

1. Introduction

Dynamic binary translation system, as a relocation tool for executable code, has been applied both in electronic commodity and in research domain for decades. The ability to manipulate the instruction stream of an executing program enabled by [1] these systems has had numerous implications in program performance, security, and portability [1]. In order to relieve the system overall overhead, researchers have attempted many algorithms or methods, where a significant improvement method| making use of code cache [26] is able to upgrade better performance. Indeed, it caches lots of

*Corresponding Author

translated or optimized code to be reused for system, yet this amortizes the cost of expensive retranslation time over the entire system program execution time. Substantively, this process increases the locality of stored code and code specialization, so program execution speed can be achieved remarkably.

Software code cache is virtually a segment of sequential memory space to store altered copies of original program instructions from low address to high address in order, which should have low overhead, good temporal locality, and minimal fragmentation. Some commercial virtual machines [14-16] still employ unbounded code cache to cache translated basic block or optimized code (i.e. superblocks) to extremely relieve extra retranslation overhead without any management overhead. In [13], Kim has explained that as the size of new huge software released grows, the size of corresponding unbounded code cache [9] will grow proportionately or exponentially. That is to say, unbounded code cache is out of state. Especially, in embedded system, the memory space must be reasonably utilized, rather than be wasted. Currently, though many dynamic binary translators, such as Strata [2], Walkabout [3], and UQDBT [22], have already employed bounded code cache [25], the utilization rate of code cache space is so low that more space cannot be fully used. In DBT system, not all of the blocks (the unit of codes called in DBT) in code cache are highly in use during a certain period of the running time, which means at this period, even if we reduce the code cache's size to the size just holding the highly-used codes, it might not influent the performance of the application too much. So the key point becomes how to decide the size of code cache at a given time.

Unlike pages in memory, blocks in code cache have unfixed sizes each other, and linking between them is also considered, making the prediction imprecise. So LRU algorithm, as the conventional replacement policy, causes so many fragmentations that more code cache space will be wasted, if it is applied on software code cache. Furthermore, the simple code cache management, like pure Flush replacement policy, is widely employed in code cache, especially leading no fragmentation. Attentively, Flush algorithm only clears all blocks when cache is full, without considering the program behavior. This causes that extra retranslation overhead would exponentially increase, that is, thrashing [5] usually rises, since some frequent-executed codes are repeatedly flushed and regenerated in code cache. So when many resource-consuming applications are running on the same physical machine, to promise high utilization rate of code cache space and system overall performance, a novel code cache management is needed.

Working set gives us another chance. In DBT system, working set means the collection of the most recently used blocks of the program in the software code cache, which reflects the program behavior as well. In this paper, we present a novel code cache management---SCC (Static Code Cache) based on working set, which outperforms Flush policy since the new one executes alternative policy according to working set. Although some performance improvement can be achieved, associated with SCC policy, extra unused code cache space is vacant. Especially, in memory constrained embedded

devices such as cellular phones, global positioning units and medical devices, vacant code cache space must be avoided. So we also propose a new dynamic replacement policy---DCC (Dynamic Code Cache) based on SCC policy. Indeed, DCC policy mainly inclines to reduce the memory requirement, enhance the utilization rate of code cache space with little or no performance sacrifice. Our job is to find the working set at different running time and accordingly adjust our code cache size to the working set's size from time to time. The code cache does have a bound but it doesn't take the full size all the time. Instead, it just takes part of it at the beginning and the size is dynamically changing, which could grow up to the bounded size at most, decided by the working sets at that time. When the transition of working set happens or the upper bound of code cache size has been reached, the flushing operation is trigged to clear code cache. We believe it would not affect the performance too much, as we take the advantage that the working sets are closely relative to the behavior of the program. In a word, it's a dynamic code cache which would adjust its size to the program behavior dynamically, saving part of the memory resource with a little sacrifice in performance. In particular, the novel contributions of our work are:

In this paper, a working-set-based replacement policy---SCC is proposed firstly. This one in the aspect of performance outperforms Flush policy, due to execution flow following program behavior. In addition, self-adjusting threshold used to decide working set is analyzed.

To efficiently utilize code cache space, another novel replacement policy---DCC based on SCC policy is presented, associated with little performance victimization that can be ignored.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Then section 3 describes the features of code cache and working set in DBT system. In addition, section 4 introduces SCC policy, the method how to judge working set in DBT system, and gives qualitative analysis of SCC. In addition, in section 5, we introduce DCC policy with several key points. Section 6 shows the experimental results about SCC and DCC, and gives the detailed analysis. Finally, we conclude the paper in section 7.

2. Related work

In this section, we will introduce some conventional dynamic translation or optimization systems how to take advantage of code cache, and discuss existing cache management utilized in various systems. Then we also simply exploit the advantage of our method relative to others.

2.1. Traditional code cache management

In 1996, dynamic binary translator---DAISY [15] developed by IBM, dynamically translates from PowerPC binaries to VLIW instruction codes. Based on this project, in 2000, BOA framework [16] presented in IBM allows PowerPC code to execute on a VLIW/EPIC processor. The Transmeta Crusoe processor [14] shipped with Code-Morphing Software (CMS) executes binary translation from IA-32 to an underlying proprietary VLIW architecture, which is the first commercial processor authentically integrated with binary translation technique. However, powerful code cache management policies are not definitely characterized in these dynamic binary translators.

Dynamo [17], as a transparent dynamic optimizer developed from HP Labs in 1999, executes on HP-UX OS providing an efficient software-management policy for code cache. It caches superblocks in code cache and takes advantage of preemptive flush mechanism as code cache policy. Virtually, this alternative management is trigged by program phase change detected. A follow-up infrastructure is DELI [18], which is a VLIW version of Dynamo developed by Hewlett-Packard in conjunction with ST Microelectronic. The code cache management employed in DELI is a special-flush cache policy controlled by user, that is, a passive flush cache policy. As the successor of Dynamo, DynamoRIO [19] is a excellent dynamic optimizer developed by Hewlett-Packard and MIT. The attractive feature of it is that it can execute on IA-32 architecture not only in Linux but also in Windows. DynamoRIO partitions the unified code cache into two independent-distinct code caches employed to cache basic code blocks and superblocks respectively, yet the superblock cache is a thread-private cache. The cache replacement management in DynamoRIO is approximate unavailable, that is, none of evictions would happen, due to unbounded code cache used to store all translated or optimized codes. Mojo [20] exploited by Microsoft, which is targeted Windows NT running on IA-32, is able to execute several large desktop applications. It also has two code caches---a thread-private basic code cache and a thread-shared trace cache, which is managed in a heavyweight manner by suspending all other threads. With regard to code cache management, each cache is subdivided into two units. For each unit, it would be flushed in special order (i.e. FIFO, LRU) when filled with codes. This leads to complicated cache management to ensure synchronism between threads due to shared code. In 2004, a novel dynamic binary translator---DigtalBridge [21] developed by Institute of Computing Technology, Chinese Academy of Science, is able to execute from X86 to MIPS infrastructure on Linux OS. Specially, its cache management differs from others, that is, several equivalent units (space size) are attained via partitioning unified cache. The cache management for DigtalBridge is deemed as a combination policy with Flush, FIFO, and LRU. But the situation where fragmentation still embarrasses overall performance, needs to be concerned as well.

Strata [2] and Walkabout [3] are research infrastructures for dynamic binary translation that are specifically designed to be retargetable. Strata has

been retargeted to run on SPARC, MIPS, and IA-32 architectures. Walkabout, which was based on UQDBT [22], has been retargeted to execute on both SPARC and IA-32 architectures. CrossBit [7], is a resourceable and retargetable DBT system with intermediate representation (IR). Until recently, it has fully or partially supported guest platforms including SimpleScalar, IA32, MIPS, SPARC, and has fully supported the IA32 host platform. Another RISC instruction sets platform host is on the plan, for instance, PowerPC and SPARC. HDTrans [23] is a simple fast Linux-based binary translator. Its simplicity speeds up its cold code translation performance and it shows competitive performance among DBT systems that do not optimize hotspots. StarDBT [24] is a multi-platform translation system that is capable of translating application level binaries on either Windows or Linux OSes. However, the code cache management policy in these systems is to flush the entire code cache when it becomes full.

2.2. Our work

As we know, the traditional replacement strategies, such as FIFO, LRU and Flush, have been widely used in operating systems. However, due to the unequal size of each block in DBT system, the traditional strategies used in code cache might encounter some problems which would not happen in OS, such as the fragmentation, and de-linking, etc, especially causing cache space waste. As a result, they may not achieve their expected performance in DBT.

In this paper, we define working set as the set of blocks that run recently. Finally, according to working set detected, DBT system can automatically adjust its code cache space, and this avenue saves more cache space for memory-consuming applications. That is to say, it enhances the utilization rate of code cache.

3. Background

3.1. Features of software code cache

Compared with physical memory, software code cache has its explicit challenges that directly impact overall system, mainly focusing on its cached code blocks.

Unfixed-sized cached codes. The significant feature of software code cache that differs from traditional hardware cache is that the size of stored codes (i.e. translated basic block or superblock) is not fixed but variable. This conduces that when replacement algorithm used in code cache takes place (i.e. LRU), fragmentation will appear in code cache. To minimize

fragmentation or even avert fragmentation, compression is able to compact fragmentation so that extra space is to be reused, but it is too expensive for system to implement it during execution. In this process, it is necessary to revise all of the branches, for each branch links one code to another congener code (In general, codes are classified into two groups| basic block and superblock). We can see that fragmentation obtained as a byproduct when some replacement algorithm being trigged, drastically affects overall performance, so avoiding fragmentation or lowering the amount of fragmentation (decreasing compression overhead)must be taken into account when selecting powerful replacement algorithm.

Linking repair. Linking is an optimizing method implemented on all the basic blocks through modifying the machine codes after they have been executed for once [8]. All the superblocks need to be linked after it created as well. The essence of linking for basic block or superblock is that inserting jump instruction into the bottom of each code block sacrifices space size to exchange less time. Through linking between code blocks, execution from one code block to another is performed in succession rather than transforming control to system to again determine the next executable code block. This leads to a better performance. However, when replacement other than flush occurs, eviction of code blocks in code cache will bring dangling linking that causes incorrect program execution. Since one code block has several incoming and outgoing linking, how to efficiently and reasonably cope with these linkings is critical for system performance. To ensure program execution correctness, it is easy to evict outgoing linking with code block being replaced relative to incoming linking. While conventional methods to settle this embarrass situation where incoming linking related to the candidate of evicted code block should be disposed immediately, is to build a back-pointer table. It stores incoming and outgoing linking information of each code block. When replacement policy is trigged, the system will firstly lookup this table to acquire the incoming linking of eviction candidate. Then these incoming linkings of the candidate code block will be evicted. In fact, this process can carry extra run-time overhead due to lookups and occupying memory space.

Retranslation overhead. Code cache miss, as a ubiquitous problem to leave a high retranslation overhead, cannot be fully avoided, yet is only attempted to minimize occurrence frequency to some extent. The high retranslation overhead results from a series of successive program execution behaviors. That is to say, this process is that storing context information about running program, regeneration of the previously cached code, copying it into code cache, updating hash table and linkings, and restoring context information about running program and transferring control. We can see that this process is so complicated that more run-time overhead will naturally appear.

In conclusion, conventional replacement scheme---LRU cannot be adequately applied on the code cache. Though Flush clears all the block stored in the code cache to avoid additional repair overhead mentioned

above, it doesn't take program behavior into account, leading to more cache misses, but with excessive cache space.

3.2. Working set in DBT system

In traditional OS, a program's working set W(*t*, T) is the set of distinct pages at time *t* among the T most recently referenced pages. Intuitively, it is the smallest subset of its pages that must reside in main memory in order that the program operates at some desired level of efficiency. The working-set principle of memory management states that a program may use a processor only if its working set is in main memory, and that no working-set page of an active program may be considered for removal from main memory [11].

In DBT system, this principle is in practice as well. In this paper, we would rather define working set as the set of blocks that run recently [10]. Taking the loop circles of the program into account, for a certain period of time, we may regard the program is running among only several blocks. During this period of time, even if we move other blocks not belonging to this set out of the code cache, the performance of the program would not drop. Now how to determine the working sets correctly has become quite important.

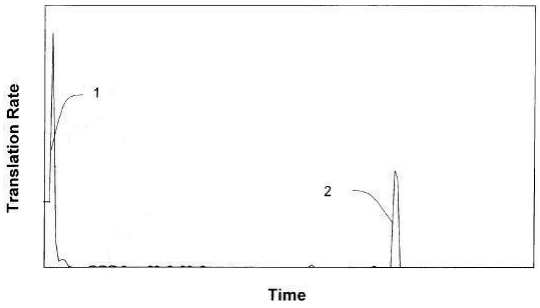


Fig. 1. Working sets detected according to translation rate

From Fig.1, we can easily see that there are 2 working sets during period of execution time, achieved by inaccurate skin-deep partitioning method. In a word, the execution flow of benchmark shows better temporal and spatial locality, and this execution process is also considered as the alternately execution of working sets.

4. Static Code Cache

In this section, we will introduce the code cache management---SCC (Static Code Cache), based on working set, which can achieve better performance than traditional replacement policy---Flush, without too much cache space. It

promises the overall performance and saves cache space, which is adequately adapt to manipulate code cache compared to conventional code cache replacement policy.

4.1. Judging criterion for working set

The theoretic. In this paper, we use the translation rate to decide the working set. Here, we define translation rate---T_{rate}:

$$T_{rate}=N_{TranslationBlock} / N_{ExecutionBlock}*100\% \quad (1)$$

In formula (1), T_{rate} represents the running program's translation rate. $N_{TranslationBlock}$ is deemed as the number of translated blocks stored in code cache, and contrarily, $N_{ExecutionBlock}$ represents the number of executed blocks.

An accompanying observation is that an increase in the rate at which translations are created---the translation rate---is often a precursor to an increase in the proportion of time spent executing within the code cache. It then follows that as the proportion of code cache execution time increases, the translation decreases. A high translation rate indicates that the translator is creating a set of translations that will be executed in the near future.

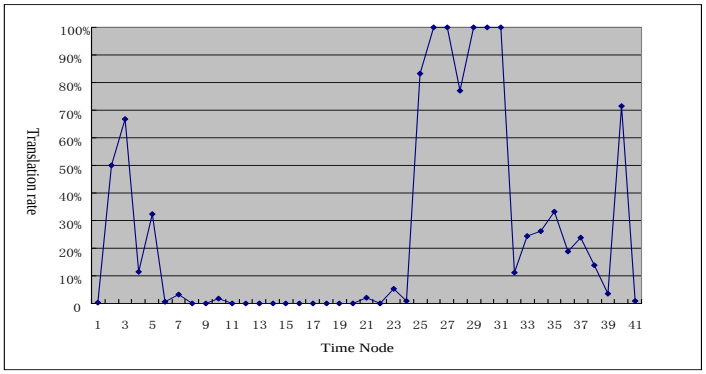


Fig. 2. Working sets detected in MCF

The set of translations can be termed the upcoming working set of the program since it represents the code that will perform the program's work in the upcoming phase of execution. A low translation rate indicates that the current working set has been captured in the cache and thus execution is occurring primarily from the cache. For a program whose behavior is characterized by the execution of different portions of code across distinct phases, the translation rate follows a regular pattern. The translation rate increases when translations for the (upcoming) working set are being

constructed, decreases and remains low as the working set executes, increases again when the next working set is constructed, and so on. This phenomenon is described in Fig.1.

In Fig.1, the rising side of the leftmost peak shows the translation rate increasing as a new working set is built in the cache. The trailing side of that peak shows the translation rate decreasing as formation of a working set nears completion. The period between the peaks shows a translation rate remaining relatively low as most of the execution occurs in the working set stored in the cache. The next peak shows that the program is entering a new phase of execution: code that has not been stored in the cache is needed for execution; another working set is being formed. This principle in general purpose program is also work, such as mcf, and this is depicted in Fig.2.

The value of thresholds. Through the description of the theoretic, we can see that the key point for judging working set in DBT system is to find the value of the two thresholds: the threshold1 and threshold2 indicated in Fig.1. These two values could determine whether we could get the right working set or not. It could be easily told that if threshold1 is set too low, the whole program might just be only one working set as a whole; if it's set too high, as threshold2 must be higher than it, the next working set might never come. Threshold2 would accordingly has the same problem. Moreover, the gap between the two values is also quite important. We have done some experiments to decide the two thresholds' value. We take one benchmark from SPECint 2000: MCF, as the test program. Firstly, we record the block ID of the first 1750 blocks of MCF. As the program is running, the same block would be executed again. That is, in a period of time, the translated code blocks stored in code cache can be reused. In Fig.2, many working sets exist, which keep to the principle mentioned above, and there are many transition points used to judge working set. Through this experiment, we can find different working sets easily, but how to divide working set accurately according to many transition points is the key problem. So according to the transition point of two working sets, we can achieve the accurate thresholds through the following experiments in Table 1.

Table 1. Transition points of working set detected with different thresholds

Threshold1	Threshold2	Working sets transition number
20	25	2
20	30	2
30	35	1
30	40	0
40	45	0
40	50	0
50	55	0
50	60	0

Since the two thresholds could be neither too high nor too low, we set the range between 0.2-0.6. On the other hand, the gap between the two

thresholds should not be too wide, so we take 5%-10% as two choices. The results based on formula (1) are shown in Table 1. In a period of time, we expect that only one working set can be detected according to Threshold1 and Threshold2. But if more than one working sets or 0 working set is detected, that illuminates that the thresholds are not competent for this system. So in this system, we choose 30% and 35% as the two thresholds.

4.2. Self-adjust thresholds

However, different programs have different behaviors; a fixed threshold could not match all the programs. As a result, we try to make our strategy adjustable to the program run on BT system, which processing as follows:

When we continually flush our code cache 10 times with fully bounded size, and meanwhile threshold1 is reached but no working set transition detected, we believe our threshold is a little bit too high for this program and we minus 2% from both two thresholds.

When we continually flush our code cache 10 times with fully bounded size, meanwhile threshold1 is never reached within these 10 times, we add 2% to both two thresholds.

This self-adjust method is so efficient that the thresholds are flexible when facing different programs.

4.3. Static Code Cache

Compared to Flush policy, SCC replacement policy is based on the transition of working set to do flush operation, rather than code cache filling with translated blocks. That is to say, Flush is a passive flush algorithm, yet SCC is a active flush algorithm. Due to different alternative condition between the two strategies, the performance of them is impacted as well. With tradition replacement policy---Flush, the code cache may store lots of unexecuted code blocks in a long time, and that will bring extra space waste. However, the novel policy---SCC can actively flush the unnecessary code blocks timely, according to program behavior, so that is advantage for the system to save more space. Then we will give the qualitative analysis about performance and space in detail.

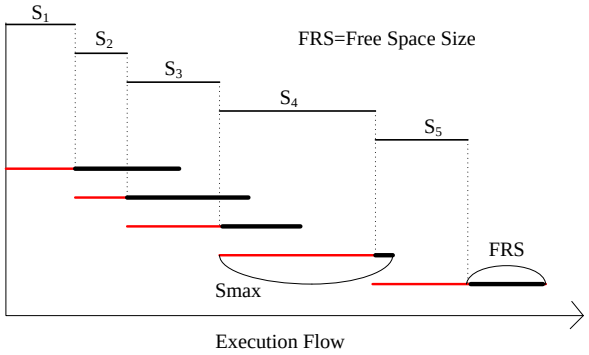


Fig. 3. Difference between Flush Policy and SCC Policy

Here, we define the size of code cache is *Smax*. Suppose that one source program is executed through 5 working sets: WS_1, WS_2, WS_3, WS_4, WS_5. The size of each working set mentioned above is respectively: S_1, S_2, S_3, S_4, S_5, where $Smax > S_i$ (i=1,2,3,4,5). And the same source program is also executed with Flush policy. The comparison of them is depicted in Fig.3. If Flush replacement policy is selected, when code cache is full of translated code blocks, the code cache will be adequately cleared without considering program behavior. That leads that more than one working set will be flushed, where there exists one working set being constructed. Indeed, some code blocks not executed in a period of time, still stored in the code cache, yet those blocks occupy so much cache space. If we choose SCC replacement policy, the active flush method is applied on code cache, rather than passive one. The flush condition is altered to working set transition, and this conforms to program behavior, which has been proved in section 4.1. So in Fig.3, extra free space (overstriking black beeline showed in Fig.3) can be saved when code cache fully flushed. Indeed, the overall space utilization with SCC policy is total of working sets' size, that is, $\sum S_i$ (i=1,2,3,4,5), and that is further smaller than 5 * *Smax* (When passive Flush policy is employed, all the space utilization is 5 * *Smax*).

5. Dynamic Code Cache

Although SCC policy can save extra free space compared to passive Flush policy, the space saved cannot reused by other applications. Consequently, we propose another novel replacement policy---DCC (Dynamic Code Cache), which gives other applications another chance to reasonably utilize extra free cache space. In addition, if most of working sets in the executed program are too small, this leads that only a few code cache space is utilized. So, to efficiently utilize code cache space compared to SCC policy, we set another parameter, S_{part}, code cache initial size in DCC algorithm, corresponding with system initial size.

5.1. Code cache initial size

On the one hand, the saved code cache space cannot utilized by other applications, associated with SCC replacement policy. On the other, as the transition of working set might happen at any time, it is quite a waste if we flush the code cache when just a little part of it (i.e. 10%, 20%) has already been used to form working set. So we propose another code cache parameter--- S_{part}, which is to take part of the code cache initial size as the initial size, and it is a good idea to make it a rule in our strategy. If the S_{part}-sized code cache has not been full-filled, we would not do the flush job. It is quite obvious that if we check whether to flush the code cache after it's been used more than a certain percent, the performance should improve for two reasons: the flush time could be reduced compared to SCC replacement policy, so the overhead of this part could be avoid; since the code cache could contain more than one working sets, the formal ones could be reused before they're cleaned, which could save the overhead of translating them again.

The percentage of space size could be neither too small nor too big. That is because the extreme situation may lead that frequently adjusting thresholds used to judge working set (too small) or more working sets simultaneously cached in code cache (too big). Here, we take the middle value as the result of this point, which is 50%.

5.2. Dynamic-size code cache

When we check whether the transition of working set happens after the S_{part}-sized code cache has been fully taken, if so, a flush job should be done; if not, what should we do? or adjust our code cache to working set? The answer could easily be found as increasing the code cache size since we did not fully take the whole size at the beginning, till the transition happens. But how much should we add as we'll never know when the transition would happen, and the increasing size of code cache S_{add} will be discussed in section 6.

After discussing the points above, our policy becomes more and more clear:

- We initialize our code cache with S_{part} of the given size in DBT system.
- As program running on, we keep on recording the translation rate---T_{rate}. If it drops below *Threshold1*, we begin to watch it whether would rise over *Threshold2*, if so, we set the flag of working set transition true.
- When the initial size is fully taken by blocks, we check the flag in the second step, if it is true, we flush code cache, start over again to record translation rate and do the second step; if not, we apply S_{add} more of the code cache size, when it is full, we do this step again, until we reach the bound of code cache size.
- If the bound of the size has been reached, we flush the code cache and restart to record translation rate.

- During these processes above, if we continually flush our code cache 10 times with fully bounded size, meanwhile *Threshold1* has never been reached, or *Threshold1* is reached but *Threshold2* is never touched, we would accordingly drop or rise our two threshold by 2%.

6. Evaluation and analysis

To evaluate this novel code cache management in DBT system, we have applied it on original DBT system---CrossBit, which is a resourceable and retargetable DBT system with IR [7]. That is also a large basic research platform, based on the CrossBit, and research works are extended, such as multi-core technique, code behavior analysis, mobile computing for Thin-client in heterogeneous resource, distributed virtual execution strategy, defense for anomaly attack, and swam intelligent, etc. In addition, SPECint 2000 [12] is selected as the test benchmark. And the configure of physical machine is that: CPU---Intel Core I5 (2.66GHz * 4), 8GB memory with Linux kernel version 2.6.33.4. The size of code cache---S_{max} is assigned about 32 KB (Since our code cache management will be extended on embedded system in future, the size of code cache assigned is so small). The traditional replacement strategies---Flush and LRU are employed to do comparison with SCC and DCC. We use the conclusion from section 5.1: using the 50% size as the initial one for code cache (S_{part} is 16 KB).

First, we do the experiment to test difference between SCC policy and Flush policy. In this experiment, the *Threshold1* and *Threshold2* are initialized as 30% and 35%, respectively. In addition, according to different programs, the thresholds will be adjusted by themselves. The results about their performance are shown in Fig.4.

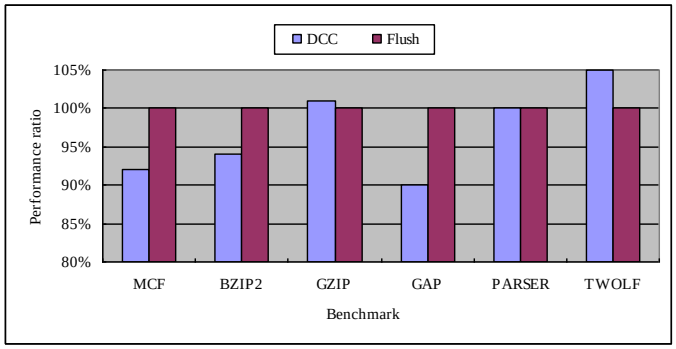


Fig. 4. Relative Runtime of Benchmarks using DCC and Flush (Normalized by Flush policy)

From Fig.4, we can see that the performance with SCC policy significantly outperforms that of Flush policy, and the average performance improvement

is about 3%. But GZIP and TWOLF benchmarks are the victims. The reason why GZIP and TWOLF undergo lower performance is that: since the thresholds used to create working set is decided by T_{rate}, the translation rate---T_{rate} is also determined by $N_{TranslationBlock}$ and $N_{ExecutionBlock}$ according to formula (1). To accurately get them, profile instructions should be added into each code blocks to record corresponding information. Here, we give the execution time of code blocks in Table 2.

Table 2. Execution time of each benckmark

Benchmark	Execution time
MCF	53011039
BZIP2	530628945
GZIP	1948235411
GAP	48558303
PARSER	238666896
TWOLF	1131632548

In Table 2, the execution time of GZIP and TWOLF are so many that the overhead caused by profile instructions is relatively increased, so the performance is lower naturally. In addition, thrashing occurring in code cache, brings inaccurate working set, leading lower performance.

Although the performance improvement is achieved associated with SCC policy, the extra free space is still vacant. So, we propose DCC policy for other applications to further efficiently utilize saved code cache space. In DCC policy, we should define the increasing grain when S_{part}-sized code cache fills with translated blocks, and this point is discussed in section 5.2. We run the six benchmarks on CrossBit with two different strategies as we increase the cache size by 5% (fine-grain), 10% and 20% (coarse-grain) each time and test their performance. Compared with the Flush strategy's running time, we get the result in Fig.5.

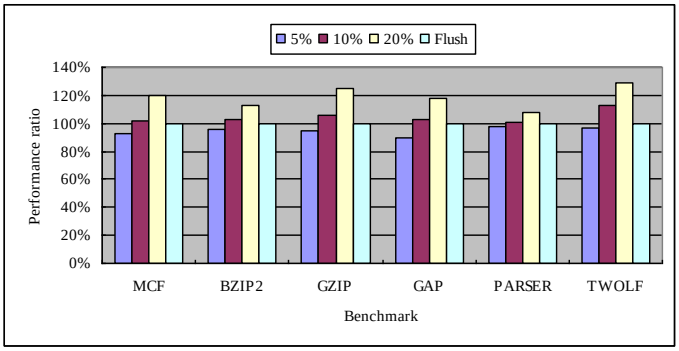


Fig. 5. The running time with different increasing grains (Normalized by Flush policy)

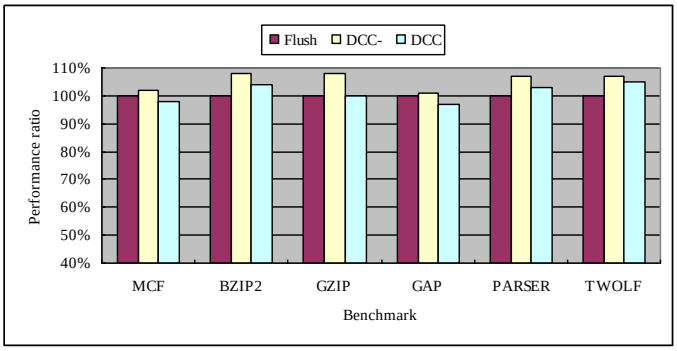


Fig. 6. The performance with different replacement policies (Normalized by Flush policy)

In Fig.5, fine-grain increasing (5%) mode is adapt to dynamic binary translation system, which is better than other increasing modes. Indeed, the remain space of code cache can be reused by other applications. Since fine-grain increasing mode can save more space for other applications, the corresponding performance outperforms that of others as well. Here, we select 5% increasing grain as the increasing mode when initial space filling with translated blocks.

Furthermore, we apply several replacement policies on the CrossBit to test the system performance, such as Flush, DCC, and DCC- (DCC- is the DCC policy without thresholds self-adjusted).

In Fig. 6, we can see that the original replacement policy (i.e. Flush) outperforms the novel one presented by us, especially DCC-. And DCC policy outperforms DCC- a little. Indeed, either DCC or DCC- spends time dynamically adjusting code cache, but the conventional replacement policy only considers space size without any adjustment on code cache. So the performance caused by DCC and DCC- is lower than the original one. The distinction performance between DCC and DCC- is mainly from the accurate thresholds used to decide working set. The more accurate working set is, the better performance improvement system can be achieved. In DCC-, invariable thresholds will cause inaccurate working set, and the reason is analyzed in section 4.2. By the way, the decreasing performance with DCC policy is about 2%-5%, compared to Flush policy, since extra code cache space occupied by other applications cannot retrieved for code cache timely. Though the new replacement policy---DCC causes some performance degradation (average of that is about 1.17%), extra code cache space is saved that can be used by other applications. In embedded system, reasonably utilizing finite memory space (code cache space) will achieve another performance improvement or promise adequate function for different applications.

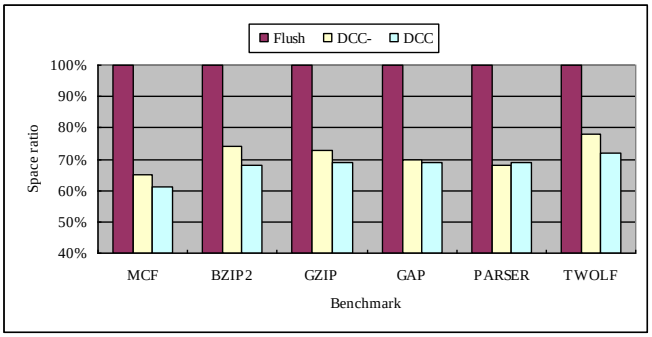


Fig. 7. The space used by different replacement policies (Normalized by Flush policy)

Next, we do the experiment to test memory space utilization. When system initializing, 32 KB memory space is assigned to code cache. Especially, the code cache with Flush replacement policy constantly occupies this section of memory, so other applications cannot preempt it. However, the novel replacement algorithms---DCC and DCC- don't occupy overall assigned memory space all the time. That is to say, the unused memory space assigned to code cache can be utilized by other applications, and it improves space utilization rate. The results is depicted in Fig.7.

Fig.7 depicts space utilization rate of each replacement policy. The simple replacement policy---Flush only considers that whether the cache space fills with translated code blocks, rather than program behavior and space utilization rate. About space utilization, the novel policies---DCC- and DCC are significantly outperform the original one, since they follow program behavior to do replacement (the replacement unit is working set). That is to say, extra free code cache space can be reused by other applications. From Fig. 7, the space utilization of DCC strategy is better than that of DCC-, since self-adjust thresholds impact the final space utilization rate. In DCC-, fixed thresholds that used to judge working set is not flexible adequately, so flushing operations with fully bounded size is more than that of DCC.

In conclusion, this new replacement policy can save extra code cache space used by other applications with a little performance improvement. As well, compared to traditional replacement policy---Flush, the novel one has more advantage, such as better performance and saving extra memory space.

7. Conclusion and future work

Code cache management in dynamic binary translation system is deemed as a crucial yet intractable issue. The high cost of preparing translated basic blocks and superblocks inserted into a code cache has incurred many researchers to slide over this serious issue through implementing either an larger code, such as merging several code blocks, or an unsophisticated

replacement scheme, such as flush policy. However, the traditional replacement policies cannot fully adapt to dynamic binary translation system. So in this paper, based on working set, a novel replacement policy---SCC is proposed. The performance of it outperforms the original one---Flush, due to it considering program behavior, but extra free code cache saved is still vacant. Then we present another new replacement policy---DCC, which is not in light of program behavior, but also make saved code cache space to be reused by other applications. Unfortunately, the system performance with DCC policy is decreased. Thus, our future work is to improve its performance, and implement our method on other dynamic binary translation systems.

Acknowledgment. This work was supported by the National Natural Science Foundation of China (Grant No. 60970108, 60970107), the Science and Technology Commission of Shanghai Municipality (Grant No. 09510701600, 10DZ1500200, 10511500102), IBM SUR Funding and IBM Research-China JP Funding. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

1. Hazelwood, K., Cohn, R.: A Cross-Architectural Interface for Code Cache Manipulation. In Proceedings of the 4th International Symposium on Code Generation and Optimization. Manhattan, New York, USA, 17-27. (2006)
2. Scott, K., Kumar, N., Velusamy, S., Childers, B., Davidson, J. W., Soffa M. L.: Retargetable and reconfigurable software dynamic translation. In Proceedings of the 1st International Symposium on Code Generation and Optimization, San Francisco, California, 36-47. (2003)
3. Cifuentes, C., Lewis, B., Ung, D.: Walkabout-A retargetable dynamic binary translation framework. In Proceedings of the 4th Workshop on Binary Translation, Charlottesville, Virginia, 22-25. (2002)
4. Ung, D., Cifuentes, C.: Machine-adaptable dynamic binary translation. In Proceedings of ACM SIGPLAN Workshop on Dynamic and Adaptive Compilation and Optimization, Boston, Massachusetts, 41-51. (2000)
5. Tannenbaum, A.: Modern Operating Systems, Second Edition. Prentice Hall, New Jersey. (2001)
6. Source codes and Introduction of CrossBit (2005). [online]. Available: http://sourceforge.net/projects/CrossBit/
7. Shi, H. H., Wang, Y., Guan, H. B., Liang, A. L.: An intermediate language level optimization framework for dynamic binary translation. ACM SIG/PLAN Notice. Vol. 42, No. 5, 3-9. (2007)
8. Hiser, J. D., Williams, D., Hu, W., Davidson, J. W., Mars, J., Childers, B. R.: Evaluating indirect branch handling mechanisms in software dynamic translation systems. In Proceedings of the 5th International Symposium on Code Generation and Optimization, San Jose, CA, USA, 61-73. (2007)
9. Bruening, D., Garnett, T., Amarasinghe, S.: An infrastructure for adaptive dynamic optimization. In Proceedings of the 1st Annual International

Symposium on Code Generation and Optimization, San Francisco, California, 265-275. (2003)
10. Banerjia, S., Bala, V., Duesterwald, E.: Preemptive replacement strategy for a cacheing dynamic traslator. USA Patent, No.US6,237,065 B1. (2001)
11. Denning, P., Schwartz, S. C.: Properties of the working-set model. Communications of the ACM. Vol. 15, 191-198. (1972)
12. SPEC CPU2000 Documentation (2010). [online]. Available: http://www.spec.org/osg/cpu2000/docs/
13. Hazelwood, K., Smith M. D.: Code Cache Management Schemes for Dynamic Optimizers. In Proceedings of the 6th Annual Workshop on Interaction between Compilers and Computer Architectures. Cambridge, MA, 102-110. (2002)
14. James, C. D., Brian, K. G., John, P. B., Richard, J., Thomas, K., Alexander, K., Jim M.: The transmeta code morphing™ software: Using speculation, recovery, and adaptive retranslation to address real-life challenges. In Proceedings of the 1st International Symposium on Code Generation and Optimization. San Francisco, California, USA, 15-24. (2003)
15. Ebcioglu, K., Altman, E. R.: DAISY: Dynamic complication for 100% architectural compatibility. In Proceedings of the 24th International Symposium on Computer Architecture. Denver, Colorado, 26-37. (1997)
16. Altman, E. R., Gschwind, M., Sathaye, S., Kosonocky, S., Bright, A., Fritts, J., Ledak, P., Appenzeller, D., Agricola, C., Filan, Z.: BOA: The architecture of a binary translation processor. IBM Research Report RC 21665. (1999)
17. Bala, V., Duesterwald, E., Banerjia, S.: Dynamo: A Transparent Runtime Optimization System.In Proceedings of the 4th International conference on Programming language design and implementation. Vancouver, British Columbia, Canada, 1-12. (2000)
18. Desoli, G., Mateev, N, Duesterwald, E., Faraboschi, P., Fisher, J. A.: Deli: A new runtime control point. In Proceedings of the 35th International Symposium on Microarchitecture. Istanbul, Turkey, 257-268. (2002)
19. Bruening, D., Duesterwald, E., Amarasinghe, S.: Design and implementation of a dynamic optimization framework for Windows. In Proceedings of the 4th ACM Workshop on Feedback-Directed and Dynamic Optimization, Austin, Texas. (2001)
20. Chen, W. K., Lerner, S., Chaiken, R., Gilles D. M.: Mojo: a dynamic optimization system. In Proceedings of the 3rd ACM Workshop on Feedback-Directed and Dynamic Optimization. Monterey, CA, 81-90. (2000)
21. Bai, T. X., Feng, X. B., Wu, C. G., Zhang, Z. Q.: Optimizing Dynamic Binary Translator in DigitalBridge. Journal of Computer Engineering, Vol.31, No.10, 103-105. (2005)
22. Ung, D., Cifuentes, C.: Machine-adaptable dynamic binary translation. In Proceedings of ACM SIGPLAN Workshop on Dynamic and Adaptive Compilation and Optimization. Boston, MA, 30-40. (2000)
23. Swaroop, S., Jonathan S. S., Prashanth P. B.: HDTrans: An Open Source, Low-Level Dynamic Instrumentation System. In Proceedings of the 2th ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, Ottawa, Ontario, Canada, 175-185. (2006)
24. Wang, C., Hu, S., Kim, H., Nair, R. S., Breternitz, M., Ying, Z. W., Wu, Y. F.: StarDBT: An Efficient Multi-platform Dynamic Binary Translation System. In Proceedings of the 12th Asia-Pacific Computer Systems Architecture Conference. Korea, 4-15. (2007)

25. Baiocchi, J. A., Childers, B. R., Davidson, J. W., Hiser, J. D.: Reducing pressure in bounded DBT code caches. In Proceedings of International conference on Compilers, architectures and synthesis for embedded systems. Atlanta, GA, USA, 109-118. (2008)
26. Bruening, D., Kiriansky, V.: Process-shared and persistent code caches. In Proceedings of the 4th ACM SIGPLAN/SIGOPS international conference on Virtual execution environments. Seattle, WA, 61-70. (2008)

Ruhui Ma is currently a Ph.D. candidate at Shanghai Jiao Tong University, China. He received the B.S. and M.S. degrees at School of Information and Engineering from Jiangnan University in 2006 and 2008, China, respectively. His main research interests are in virtual machines, computer architecture and compiling.

Haibing Guan received his Ph.D. degree in computer science from the Tongji University (China), in 1999. He is currently a professor with the Faculty of Computer Science, Shanghai Jiao Tong University, Shanghai, China. He is a member of CCF. His current research interests include, but are not limited to, computer architecture, compiling, virtualization and hardware/software co-design.

ErZhou Zhu is currently a Ph.D. candidate at Shanghai Jiao Tong University, China. He received the M.S. and B.S. degrees in computer science and technology in Anhui University, China, in 2004 and 2008 respectively. His research interests include virtual machine, binary translation and computer architecture.

Yongqiang Gao is currently a Ph.D. student of at the Shanghai Jiao Tong University (SJTU), China. He received his M.Sc. in Computer Applied Technology in 2009 from Northeastern University, China. His main research interests are computer architecture, virtualization and green computing

Alei Liang received his Ph.D. degree in computer science from Shanghai Jiao Tong University, China, in 2002. He is currently an assistant professor with the Faculty of Computer Science, Shanghai Jiao Tong University, Shanghai, China. His current research interests include, but are not limited to, parallel computing via swarm intelligence and virtualization computing with dynamic binary translation.

Received: March 27, 2010; Accepted: January 14, 2011.

DOI: 10.2298/CSIS101011023K

Data Extraction and Annotation Based on Domain-specific Ontology Evolution for Deep Web

Kerui Chen[1, 2], Wanli Zuo*[1], Fengling He[1], Yongheng Chen[1] and Ying Wang[1]

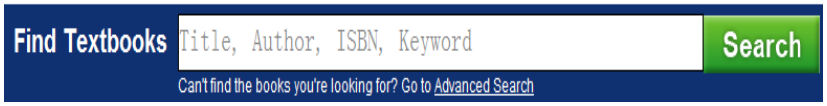
[1] College of Computer Science and Technology, Jilin University, Changchun 130012, China
Key Laboratory of Symbol Computation and Knowledge Engineering of the Ministry of Education
[2] College of Computer and Information Engineering,
Henan University of Economics and Law, Zhengzhou 450002, China
*Corresponding Author: Wanli Zuo wanli@jlu.edu.cn

Abstract. Deep web respond to a user query result records encoded in HTML files. Data extraction and data annotation, which are important for many applications, extracts and annotates the record from the HTML pages. We proposed an domain-specific ontology based data extraction and annotation technique; we first construct mini-ontology for specific domain according to information of query interface and query result pages; then, use constructed mini-ontology for identifying data areas and mapping data annotations in data extraction; in order to adapt to new sample set, mini-ontology will evolve dynamically based on data extraction and data annotation. Experimental results demonstrate that this method has higher precision and recall in data extraction and data annotation.

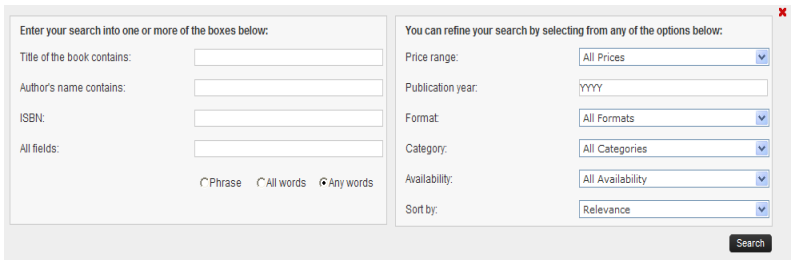
Keywords: Deep Web, Data Extraction, Data Annotation, Domain Ontology, Ontology Evolution.

1. Introduction

Currently, more and more researchers start focusing on how to manage data information hided in back-end database more effectively. As a result, according to the distribution of Web and storage depth of information, Web has been classified as “Surface Web” and “Deep Web” (Hidden Web/Invisible Web). Surface Web represents internet resources that are linked by hyperlink, such as picture, file, and static web page, and they usually could be accessed by using these hyperlinks; on the other side, Deep Web is constructed by back-end databases, and their contents are stored in relational databases of back-end web sites; unlike Surface Web, there are no hyperlinks to these web sites, it rather dynamically produces web sites contained query result records by back-end server based on query conditions.



(a) Simple of Query interface



(b) Advanced Query interface

(c) Query result display

Fig. 1. The sample of Query interface and Query result

From Figure 1, we could figure out that (a) and (b) represents two query interfaces. After inputs keyword “c program”, the server will get query result (c). The main task in this paper is to annotate each data item in (c), and to integrate results returned by various data sources into one table.

Through automatic data extraction in Deep Web, with data integration by data annotation, it would be able to provide better service to various commercial web sites, such as the seller or agency of internet commercial information; in addition, it also helps portals to provide more professional and personalized information search service.

2. Related work

In recent years, there are more and more web data extraction tools coming out, and they could be classified by functions and features into categories as follows: Languages for Wrapper Development [1-3], HTML-aware Tools [4-5], NLP based(natural language processing)[6-7], wrapper induction based[8-9], data modeling based[10-11], visual information based[12-13], and ontology based web extraction method[14].

The research on web data extraction is comparatively mature, and there are many data extraction methods available from theory to practice; on the other side, the research on data annotation is still in the infant stage both domestically and internationally. There are mainly three types of semantic annotation method targeted at data: mode based system [15-16], machine learning based [17-18], and ontology based method [19-21]. For data annotation, ontology based method is the main streaming. There are more and more researchers start to accept practicability of using ontology in data annotation, and actually make some achievements.

Multiple annotation tools mentioned above adopts heuristic rules for annotating. If only a specific Deep Web database needs to be annotated, it will be possible to use machine learning algorithm to train in sample training set; once semantic relationship between data was obtained, it would dig out a series of rule sets and apply them in annotating new web sites. Although this method is only available in specific Deep Web pages, it does not work for other Deep Web pages in the same domain; therefore, simply use machine learning method cannot suit a mass of isomerous Deep Web pages.

While some annotation tools did use ontology, many problems still existed even they suited for annotation of multiple domains. For example, majority of annotation tools are targeting at pure text, so their ontology definitions are relatively complex, less efficient, and not adapting to Deep Web's structural data features. Moreover, as most defined ontology are static, precision and adaptability needs to be improved.

The organization structure of this paper is listed as below: the third section discussed construction of mini-ontology; then, the following section detailedly introduced how to use ontology information to identify data records area; based on predefined division and alignment rule, data in data records area would be divided and aligned`; in this way, each data record would be extracted out; the fifth section mainly focused on using ontology to annotate extracted data records; finally, in sixth section, we suggested a evolution frame to examine how mini-ontology evolved with data extraction and annotation; The last two sections evaluate experiments and provide a future outlook.

3. Mini-ontology generation

3.1. Definition

PVAs(Programmer Viewpoint Attributes): attributes extracted from query interface in the view of web programming. These PVAs extracted from HTML labels were similar to values of following labels: <label>, <input>, <option>, <select> and so on.

UVAs (User Viewpoint Attributes): attributes extracted from query interface in the view of web users. Normally, values of UVAs follow the label closely.

3.2. Attribute recognition

The extraction of query interface's attributes took example form Automatic Attribute Extraction method proposed by YOO JUNG AN et.al [22], there were two steps for extracting attributes from query interfaces:

(1)To acquire PVAs from source code of query interfaces;

The extraction process of PVAs is executed in the order as follows:

Step 1: to extract all string sets SS from Html document's tags, and to record keysets between label <label> and </label>, as well as label <Select> and </Select>, in KW; meanwhile, duplicate ones will be removed;

Step 2: to traverse each string in SS for identifying whether special symbols like ":"、"/"、"{"、"}"、"#"、"$"、"&"、"*"、">"、"+"、"\"、"="、"?"、"<"、"["、"]"、"@"、"_" were included; if so, the string would be divided into two sub-strings by using symbols above as dividing line;

Step 3: to traverse each string in SS for identifying whether capitalized letters were included; and then use capitalized letters as dividing line to divide string into two sub-strings;

Step 4: to traverse each string in SS for identifying whether key words in set KW were included; if so, this query key word would be used as boundary for dividing string into two sub-strings;

Step 5: to record NSS, the number of times that string SSi has been divided;

Step 6: to recombine sub-string SSi, divided from each string; while only neighboring sub-strings could be combined, the number starts from 2 until its value reaches NSSi, and combined strings would be stored in SSi.

Step 7: To recheck string set SS and to delete duplicated strings;

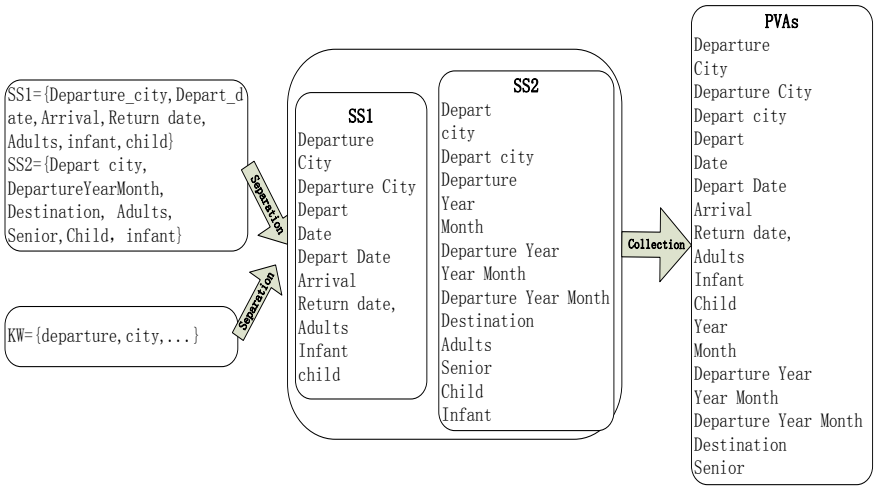


Fig. 2. Sample process of acquiring PVAs

Shown as fig.2, there were two original string sets SS1 and SS2 extracted from query interface's HTML labels, and key word set KW extracted between <label> and </label>, and <Select> and </Select>.

We first divide SS1; since "Departure city" and "Depart date" both contain special symbol "_", they should be divided into four strings: "Departure", "city", "Depart" and "date". Right after the division, the combining process will start. For instances, "Departure" and "city" could be combined as "Departure city", while "Depart" and "date" could be combined as 'Depart date"; the final output set is showing as SS1 in the second circle.

Then, we start dividing SS2. As "Departure city" contains "city" from key word set KW, it needs to be divided. Similarly, "DepartureYearMonth" would be divided into "Departure", "Year", and "Month" as it contains capitalized letter "Y" and "M". After division, combining process will start again. "Departure", "Year" and "Month" could be combined as "Departure Year", "Year Month" and "DepartureYearMonth"; the final output set is showing as SS2 in the second circle.

Finally, we combine SS1 and SS2, and pick out duplicated string once checked out, such as "Departure", "Depart", "city" and so on; after that, PVAs set would be obtained.

(2)To acquire UVAs from texts of query interfaces;

The steps of extracting UVAs from query interfaces are listed below:

Step 1: to traverse HTML source codes for searching start tag <Option> and end tag </Option>; then, to remove all free texts between these two tags.

Step 2: to traverse HTML source code again, and to save free texts between any two labels in set UVAs in the form of strings.

Step 3: to remove duplicated texts in UVAs.

Step 4: to traverse all strings in set UVAs.

When there is a string that contains special symbols, the special symbol will be used as dividing line to divide that string into two sub-strings, which will be stored in set UVAs instead of previous string.

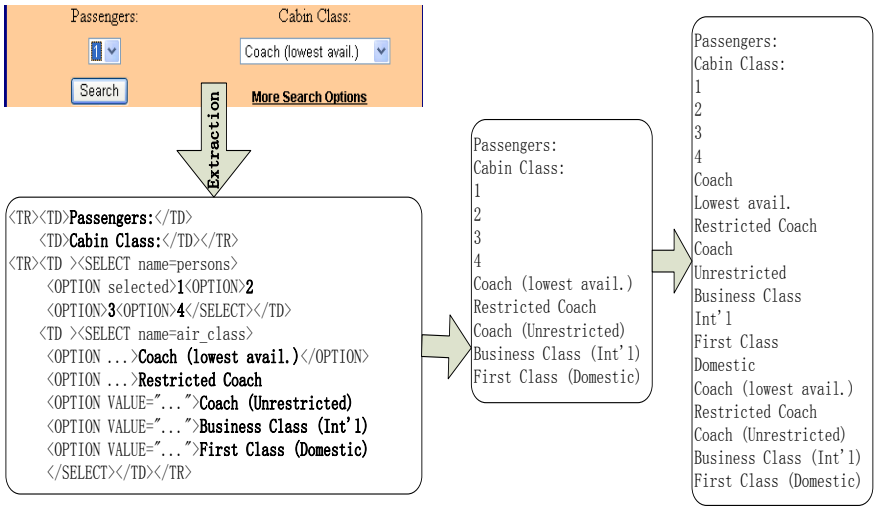


Fig. 3. Sample process of acquiring UVAs

Shown as fig .3, UVAs is the instance extracted from labels, such as the free texts between <TD> and </TD>: “Passengers:” and “Cabin Class:”, and free texts between <Option> and </Option>: “1”, “2”, “Coach (lowest avail.)”, “Restricted Coach” and so on. The instance value needs to be judged whether it contains special symbols in order to divide. Since “Coach (lowest avail.)” contains “(“, it will be divided into “Coach” and “lowest avail.”. Then, the set UVAs will be obtained.

3.3. Mini-ontology generation

Not only does construction information of Mini-ontology include attribute information of query interface, but it also contains abundant instance information in query result pages. The extraction of query result data instance and construction of ontology adopts method in [23].

4. Automatic data extraction

Currently, data extraction is facing challenges as below:

(a) Data have different layouts and patterns in web sites;

(b) It’s more difficult to dig right relevancy between labels as a label in the table was constructed by cells with multiple levels;

(c) Data items use different expression forms;

(d) There were isolated information and those may cause ambiguity in the table;

(e) It's hard to identify and extract when single data record existed in some web sites.

In order to solve problems mentioned above, this paper suggests an ontology based extraction method, which had been confirmed in experiment for its effectiveness of extracting data in web sites.

In fig .4, the whole process of data extraction has been demonstrated; the first step is to convert query result pages to DOM trees; then, to identify interested data areas by using ontology's label classifier and instance classifier; the third step is to divide data records based on the predefined heuristic rules; finally, to align extracted data records.

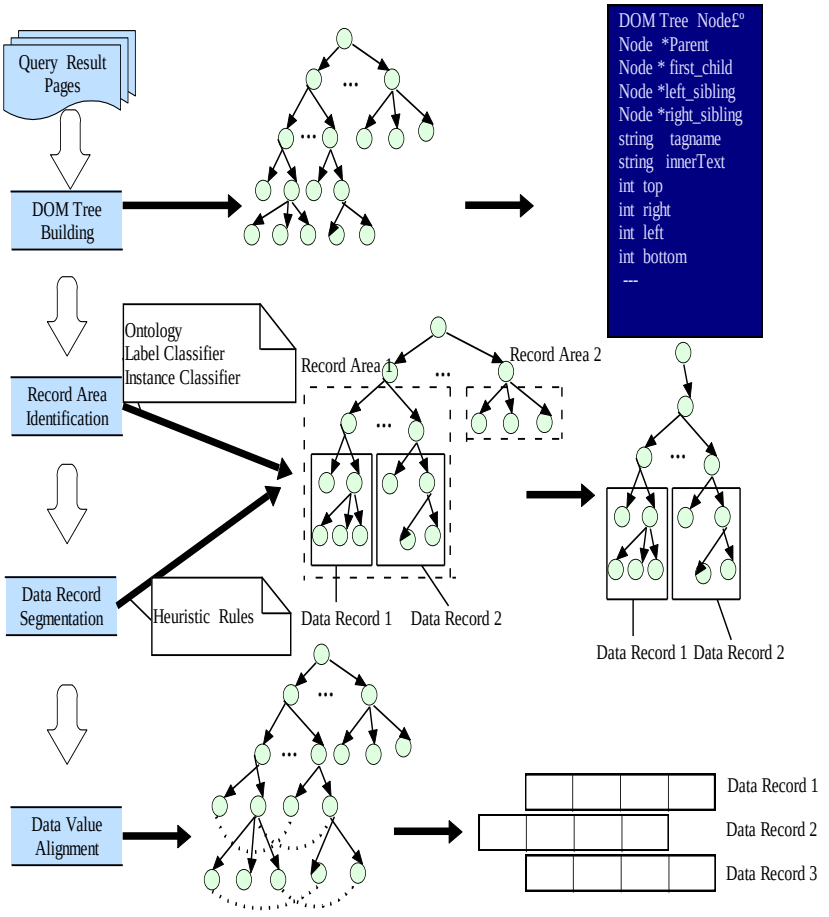


Fig. 4. Data extraction procedure

4.1. DOM tree building

Usually, a tag contains a pair of switch symbols, such as “<” and “>”, and a tag also contains a set of tag attributes. This paper designs tags in two types: start tag and end tag. As these tags are paired, the only difference is the extra symbol “/” in the beginning of end tag. For example, in figure 3.2, <table> is the start tag while </table> is the end tag. In the process, there were two types of tags that will not be considered: the first is the tag begins with “<!”, while another one is the end tag that has no associated start tag.

For extraction of query result pages instances, we first convert result pages to DOM trees. The construction method used in this paper is to use tags with visual aids, which can help deduce structural relationship between tags and construct stronger DOM tree.

The construction steps were shown as follows:

(1) To use browser’s rendering engine to find out four boundary values of each element in HTML;

(2) To check and construct from first tag in the source code of documents.

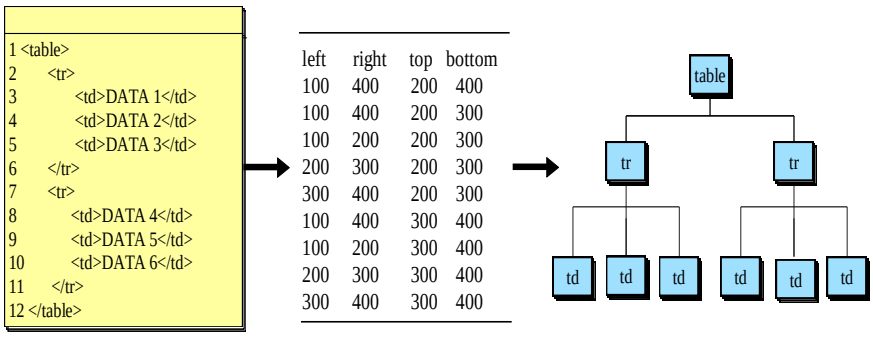


Fig. 5. HTML encoding sample, boundary coordinate and tag tree structure

Fig.5 demonstrates the construction process of DOM tree; first, the analysis chart provides source code for a table with 2 rows and 3 columns; then, it uses browser’s rendering engine to acquire boundary values of tag <table>, <tr>, and <td>; for example, the four boundary values of tag <table> are left: 100, right: 400, top: 200, and bottom: 400; finally, the tree structure of tag will be constructed according to four boundary values. Generally, browser’s rendering engine has a very high fault tolerance capability, and therefore the encoding mistakes existed in source code could be identified correctly and those constructed boundary coordinates have higher accuracy rate.

4.2. Data area identification

Although we can obtain multiple data areas, there were only one or several data areas we are interested in; hence, we need to further dig out interested data areas.

In the process of mining interested data areas, this paper synthetically utilizes methods of ontology and heuristic rules.

Heuristic rules are based on the observations as below:

(a) A group of data records containing similar target sets usually appear in the neighboring areas of web sites; in addition, they also have similar HTML tags.

(b) A data area's data records list is made up by multiple subtrees of the same father node. In other words, these data records could be considered as multiple subtrees while they have the same father node.

(c) Learning through observation, abundant data records areas usually locate at the centre of web site.

(d) If the depth of leaf node in DOM tree was too low, the node would be determined to be useless.

(e) If the data volume of node's neighboring area was too low, this node would also be useless.

The ontology used for mining data areas is based on two observations as below [30]:

(a) Data areas contain plentiful ontology information.

(b) Attributes and instances contained in ontology are usually located closely in data areas.

(1)Ontology instance relevancy:

For a given ontology O, instance set $D=\{d_1,d_2,...,d_n\}$, which comes from data records; for any original string d_i, its weight was represented as w_i; if $d_i = A_i.N$ or $d_i = A_i.FA_i$, which means d_i is the name or alias of ontology O's attribute A_i, the value of w_i will be the probability that attribute A_i appears. If d_i is the value of multiple attributes, then the value of w_i will equals to the highest probability for attribute's appearance.

In order to calculate relevancy between D and O, the following formula will be used:

$$Corr(D,O)=\frac{\sum_{i=1}^{n} w_i}{\sqrt{mn}} \tag{1}$$

In the formula, m represents the number of attributes contained in ontology O, while n represents the number of data items in the instance set.

(2) Example of data extraction algorithm

```
SearchDataRegion(R,O)
  { T ←Ri∈R;
     While(T is not leaf node)
     begin Ti←one child node of T has the largest
correlation with O;
          if(Corr(T,O)<Corr(Ti,O)) T←Ti
          else break
     End
    While(1)
    Begin Tl←the left sibling of T
          If(Corr(T,O)<Corr(Tl+T,O)) T←Ti+T
          else break
    End
    While(1)
    Begin Tr←the right sibling of T
          If(Corr(T,O)<Corr(Tr+T,O)) T←Tr+T
          else break
    End
    Return T

}
```

The algorithm mentioned above is mainly used for searching subtree that has maximum relevancy with given ontology, and hence concludes that the subtree actually is the data area we are looking for. DT is the set of data areas divided by web sites; only one subtree of the set will be picked up each time; moreover, starts from root node of that subtree, by following the order of top-down, left to right for nodes in the same level, to identify an subtree with highest relevancy.

4.3. Data record segmentation

If a data area only contains one data record, it will not be necessary to divide records; to the contrary, if it contains multiple data records, we will need to seek for an algorithm for executing division of data records.

By observing structural characteristics and encoding characteristics of data records in data areas, we propose a comprehensive division algorithm, which can utilize characteristics of tags effectively.

(1) Statistics the max number label

For candidate tags in the nodes of sub tree that has greatest Fan out of they would be ranked in an inverted order according to the number of appearance; as a final product, a annotation serial will be obtained.

(2) Identify division tag

For the layout of documents contained multiple data records, we tend to use few division tags to divide those records, such as tags: <hr>, <b>,
 and so on. Based on this observation, we can track these division tags for inspecting its frequency of being used.

(3)Standard deviation

We calculate standard deviation between each candidate tag.

(4)Repeat mode of tags

In a data record area T=ABABABA, as A and B represent two different types of tag structure, we can find two repeat mode: AB and BA. If there is only one repeat mode, then we will be able to determine the repeat mode of data record area; otherwise, we need to choose one. At this time, we can use visual aid as gap between two data records is usually greater than the one between data items in the same data record; by applying this limit, we can delete useless repeat modes.

Based on the four heuristic rules discussed above, we can divide data records in data records areas and align data records.

4.4. Data value alignment

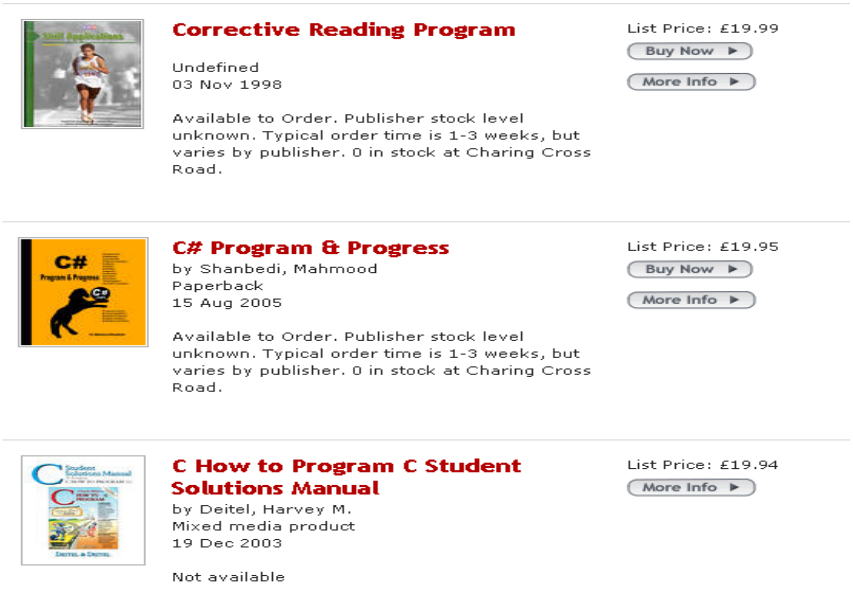


Fig. 6. Query result instance sample

Fig.6 shows a real data record sample, which can be divided into 3 data records. Table 1 demonstrates result of the three data records' alignment; from observation, we learned that numbers of nodes contained in these three data records are different. Unless the formwork of Deep Web was produced

strictly, the case mentioned above would often appear. Hence, after the data division, we still need to align divided data records.

Table 1. Fig. 6 alignment of data records

Corrective Reading program		Undefined	03 Nov 1998	Available to order….
C# Program & Progress	by Shanbedi, Mathmood	Paperback	15 Aug 2005	Available to order….
C How to Program C Student Solutions Manual	By Deitel, Harvey M	Mixed media product	19 Dec 2003	Not available

For alignment of data records in this paper, the partial tree alignment algorithm [24] has been used for aligning multiple data records produced by same data source. The main idea of this algorithm is to create an incremental seed tree for aligning multiple trees. If we consider a data record as a seed tree, a data area will have subtrees same with the number of data records contained.

5. Automatic data annotation

After the extraction of Deep Web and identification of duplicated records mentioned above, we can get an instance set $I = \{i_1, i_2, ..., i_n\}$, $\forall i_m \in I$, $i_m = (l_m, d_m)$; among it, l_m represents a tag, whose value could be null; while d_m represents an instance value, $d_m \in D$. Since d_m appears in various web sites, it's possible to have different l_m ; the annotation of data is to annotate a unified tag for d_m , which would help integrate data.

There were two types of treatments for l_m : when l_m is not null, we will adopt method suggested in [25]; when it's null, which means the attribute value is original and has no page tags, there will be two methods of annotating; in addition, the results of annotation will be stored in l_m and mapping relationship between l_m and ontology O will also be recorded eventually.

The first one adopts "Query Reset Strategy" proposed in reference [21], which was created based on observations as below: the more appropriate query conditions has been chosen, the more query result information would be returned by Deep Web background server.

The second one is based on the K-beam search algorithm of KBFS. Not only does this algorithm utilize prediction method of prediction model based on maximum information entropy models, but it also takes advantage of KBFS search algorithm's ability of seeking for optimal path. The construction steps are listed as below:

Based on maximum information entropy models, we constructed a prediction model for annotation prediction of d_m in instance set $I = \{i_1, i_2, ..., i_n\}$, $\forall i_m \in I$, $i_m = (l_m, d_m)$. The construction of this model used maximum possibility attribute model [26] as reference.

For a given instance value d_m , h is the characteristic of d_m when it appeared previously, the possibility that each attribute A_i in attribute set A contained in ontology O will be annotated by d_m is determined by prediction model as follows:

$$p(h, A_i) = \pi \prod_{j=1}^{k} \lambda_j^{f(h, A_i)} \tag{2}$$

In the formula, π is a constant, $f(h, A_i)$ is the characteristic function, and its value is either 0 or 1, which is also the weight λ_j of each characteristic in characteristic set. The possibility that instance value d_m will be annotated as attribute A_m is calculated as below:

$$p(A_m \mid h) = \frac{p(h, A_m)}{\sum_{A_i \in A} p(h, A_i)} \tag{3}$$

While $\sum_{A_i \in A} p(h, A_i)$ represents the sum of probability for all attributes.

Hence, for an original data record's data item set { $d_1, d_2, ..., d_m$ }, the conditional probability that annotated attribute tag order is { $A_1, A_2, ..., A_m$ } will be:

$$p(A_1, A_2, ..., A_m \mid d_1, d_2, ..., d_m) = \prod_{i=1}^{m} p(A_i \mid h_i) \tag{4}$$

By using calculation of prediction models mentioned above, we can get multiple prediction values; once we construct a tree based on those values, it's time to use KBFS search algorithm[27] to seek for a optimal path in prediction values' constructed tree.

6. Ontology evolution

Ontology evolution could be considered as improving ontology for adjusting to new sample sets or user requests without violating ontology's compatibility principle.

(1) Frame of ontology evolution

The frame of ontology evolution in this paper was inspired by reference [28], and it mainly has four stages as below:

(a) Capturing change information

The procedure of ontology evolution starts from capturing change information, which has two types: change based on structure, and change based on data; moreover, change based on structure requires modifying the structure of ontology, while change based on data only needs to modify associated data.

(b) Expressing change information

In order to handle captured change information, there is a further need to identify and express them in proper form; in other words, various types of change information all need related expressing method.

(c) Semantic change

Before the execution of ontology evolution, it's necessary to check whether there is a semantic change existed. The dependency between attributes in the ontology needs to be checked carefully, as they may cause ambiguous problems.

(d) Execution of ontology evolution

In this stage, the main work is to send a request of modifying ontology; once the modification has been executed, it will be recorded in order to achieve operation's reversibility.

Next, there will be detailed explanations for each stage.

(2) Capturing information change stage

Each time, after the extraction and annotation, the system will return some information to ontology evolution module. This information, including extracted instances that have not been annotated yet, will be annotated as one attribute of ontology; however, that ontology does not contain that instance value, or instances with unsure annotation; in this case, ontology evolution module needs to classifier the information based on actual situation, in order to execute expressing change information in next stage.

(3) Expressing change information stage

In this stage, classified change information would be expressed respectively. If instance that has not been annotated requires to construct a new attribute model manually for it, or ontology does not contain instance value which has been annotated as an attribute of ontology, there will be a need to create an relationship between attributes of instance and annotation; in case there are instances whose annotation were uncertain, the uncertain attribute tags should be recorded.

(4) Checking semantic change stage

Before execution of ontology evolution, there will be a semantic check for ontology operation that will be executed; for example, to check whether new

attribute model has synonymous or dependence with current attributes of ontology.

(5) Ontology evolution execution stage

Based on the special tags made by ontology during annotation, ontology evolution needs to follow rules as below:

Rule 1: when deals with data of mapping relationship between N_m and l_m, the location of ontology attribute A_m will be searched, and d_m, in the instance $i_m = (l_m, d_m)$, will be added to V_m.

Rule 2: when deals with data of mapping relationship between A_m and l_m, the location of ontology attribute A_m will be searched, and l_m, in the instance $i_m = (l_m, d_m)$, will be added to FA_m.

Rule 3: when deals with data of mapping relationship between l_m and ontology O, then a new attribute A_x will be created according to the instance $i_m = (l_m, d_m)$, and l_m will be the value of name entry N in attribute A_x tuple while d_m will be the value of attribute entry V.

As new ontology will be created after the evolution each time, multiple ontology versions will be created with the multiple evolution; hence, it's necessary to manage versions in order to prohibit losing data and to maintain ontology's consistence effectively.

7. Empirical evaluations

Test sample selected three domains from UIUC concentrating storehouse[29]: Automobile, Airfares, and Books; in each domain, 10 query interfaces has been chosen, and been provide query conditions randomly typed; then, those returned query result pages would be collected manually; in case that one query condition returned multiple query results, only first page would be selected while others would be ignored.

We manually classify collected web sites in accordance with three categories listed as below:

(a) Web site contained multiple data records (MRP)

(b) Web site contained one data records (SRP)

(c) Web site did not contain data records (ERP)

7.1. DataSet

There were totally 60 web sites collected from three domains: Automobile, Airfares, and Books, and the statistics of data records distribution in these web sites were shown as table 2. Since the typed query conditions were comparatively appropriate, there were less SRP and ERP in the collection. Moreover, there were more data records returned from Airfares domain, while

less from Automobile domain; this phenomenon was actually determined by features of domain.

Table 2. Statistics of data set

Domain	Num(Web pages)	Num(MRP)	Num(SRP)	Num(ERP)
Automobile	20	120	37	43
Airfares	20	175	9	16
Books	20	162	11	27
Total	60	457	57	86

7.2. Performance of ontology based data area identification algorithm

The performance of ontology based data area identification was shown as table 3, Airfares domain reached 100% for identifying MRP, while Books domain has lowest score, which was only 89.4%. From this, we can conclude that ontology has highest identification efficiency in Airfares domain. Books domain had lowest scores for MRP, SRP, and ERP, which could be explained by the relatively complex query interfaces in this domain.

Table 3. Performance of ontology based data area identification algorithm

Domain	MRP	SRP	ERP
Automobile	91.2%	89.7%	85.9%
Airfares	100%	96.5%	94.2%
Books	89.4%	84.4%	83.1%
Average	93.5%	90.2%	87.7%

The main idea of ontology based data area extraction method is that areas contained massive ontology information is more likely where query results located. As Books domain contained a lot of books recommendation information, which usually have complete description, they could disturb real query result records very easily, and this disturbance would be more severe when there were only a few query result records; therefore, we can find that the precision of Books domain for ERP was only 83.1% in the table.

7.3. Performance of data extraction and annotation after ontology evolution

In newly collected test sample set, the comparison for performance of ontology evolution based data extraction was shown as Table 4. We can easily find that newly evolved ontology performed better than original ontology in each domain, especially for Books domain, whose recall has been

increased by 0.9%, although Automatic domain did not increase drastically. These three domains increased 0.17% for their precisions, and 0.7% for recall. In this way, experimental results proved that improved ontology performed better in data extraction.

Table 4. performance comparison of data extraction in new training set

	Mini-ontology		Ontology after evolution	
Domain	Precision	Recall	Precision	Recall
Automobile	80.2%	78.1%	80.3%	78.5%
Airfares	92.9%	92.8%	93.0%	93.6%
Books	86.8%	83.9%	87.1%	84.8%
Average	86.63%	84.93%	86.80%	85.63%

Table 5. performance comparison of data annotation in new training set

	Mini-ontology		Ontology after evolution	
Domain	Precision	Recall	Precision	Recall
Automobile	85.0%	84.7%	85.2%	85.3%
Airfares	97.1%	95.6%	97.1%	95.7%
Books	86.3%	84.9%	86.7%	85.7%
Average	89.47%	88.40%	89.67%	88.90%

In newly collected test sample set, the comparison for performance of ontology evolution based data annotation was shown as Table 5. Just as expected, newly evolved ontology performed better than original ontology and there were increase in each domain with varying degrees. The increase in Airfares domain was not drastic as there were few attributes in this domain and relationships between attributes were also quite simple. Hence, the evolution degree of ontology in Airfares domain was not obvious, and this had little effect on annotation. In Books domain, the increase is relatively high; actually, precision and recall was increased by 0.4% and 0.8%, respectively. Experimental results demonstrated that newly evolved ontology performed better in annotating data, compared to original ontology.

8. Conclusion and future work

This paper takes full advantage of ontology's semantic information; in the process of identifying query result data areas, with the guidance of ontology, the identification will be accurate; in addition, it also solved dependence problems on web site structures existed in traditional template based method. This paper also systemically and deeply investigates design of ontology module adjusting to structural web sites, automatic construction of domain

ontology, extraction of data records in query result pages, annotation of data records, as well as the evolution mechanism of ontology.

Regarding research works discussed in this paper, there are still a lot of things need to be improved. For example, the construction process of ontology could be improved, in order to solve some ambiguous problems caused by attributes matching; furthermore, an ontology evolution evaluation mechanism also needs to be set up for controlling the increasing ontology data volume, and thereby seek for a balance point between efficiency and accuracy.

Acknowledgment. This work is supported by the National Natural Science Foundation of China under Grant No.60973040; the National Natural Science Foundation of China under Grant No.60903098; the Science and Technology Development Program of Jilin Province of China under Grant No. 20070533; the Specialized Research Foundation for the Doctoral Program of Higher Education of China under Grant No.200801830021; the Basic Scientific Research Foundation for the Interdisciplinary Research and Innovation Project of Jilin University under Grant No.450060445161;.the Basic Scientific Research Foundation for Young Teachers Innovation Project of Jilin University under Grant No.450060441075.

References

1. Crescenzi V, Mecca G.: Grammars have exceptions. Information Systems, Vol. 23, No. 8, 539-565. (1998)
2. Hammer J, Mchugh J, Garcia-Molina H.: Semistructured data: The TSIMMIS experience. In proceedings of the first east-european symposium on advances in databases and information systems, 1-8. (1997)
3. Arocena G.O, Mendelzon A.O.: WebOQL: Restructuring documents, databases and webs. In proceedings of the 14th international conference on data engineering, 24-33. (1998)
4. Crescenzi V, Mecca G, Merialdo P.: Roadrunner: Towards automatic data extraction from large Web sites. In proceedings of the 26th International Conference on Very Large Databases, 109-118. (2001)
5. Baumgartner R, Flesca S, Gottlob G.: Visual Web information extraction with Lixto. In proceedings of the 26th International Conference on Very Large Databases, 119-128. (2001)
6. Califf M.E, Mooney R.J.: Relational Learning of Pattern-Match Rules for Information Extraction. In proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence, 328-334. (1999)
7. Freitag D.: Machine Learning for Information Extraction in Informal Domains. Machine Learning, Vol. 39, No.2-3, 169-202. (2000)
8. Kushmerick N.: Wrapper induction: Efficiency and expressiveness. Artificial Intelligence Journal, Vol. 118, No. 1-2, 15-68. (2000)

9. Muslea I, Minton S, Knoblock C.: Hierarchical wrapper induction for semistructured information sources. Autonomous Agents and Multi-Agent Systems, Vol. 4, No. 1-2, 93-114. (2001)
10. Adelberg B.: NoDoSE-A tool for semi-automatically extracting structured and semistructured data from text documents. In proceeding of the ACM SIGMOD international Conference on Management of Data, 283-294. (1998)
11. Laender A H F, Ribeiro-Neto B, Da Silva A S.: DEByE-Data Extraction By Example. Data and Knowledge Engineering, Vol. 40, No. 2, 121-154. (2002)
12. Deng Cai, Yu Shipeng, Wen Jinrong, et al.: VIPS: A Vision-Based PageSegmentation Algorithm. Microsoft Technical Report, MSR-TR-203-79. (2003)
13. Liu W, Meng X, Meng W.: ViDE: A Vision-Based Approach for Deep Web Data Extraction. Knowledge and Data Engineering, Vol. 22, No. 3, 447-460. (2010)
14. Data Extraction Research Group. http://www.deg.byu.edu/.
15. S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, K.S. Mccurley, S. Rajagopalan, A. Tomkins.: A case for automated large-scale semantic annotation. Journal of Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 1, No. 1, 115-132. (2003)
16. B. Popov, A. Kiryakov, D. Ognyanoff, D. Manov, A. Kirilov.: KIM - a semantic platform for information extraction and retrieval. Natural Language Engineering, Vol. 10, No. 3-4, 375-392. (2004)
17. L. Arlotta, V. Crescenzi, G. Mecca, P. Merialdo.: Automatic Annotation of Data Extracted from Large Web Sites. In proceedings of Sixth International Workshop on the Web and Databases (WebDB 2003), 7-12. (2003)
18. J. Wang F.H. Lochovsky.: Data Extraction and Label Assignment for Web Databases. In proceedings of the 12th international conference on World Wide Web, New York, USA, 187-196. (2003)
19. Ciravegna F, Dingli A.: User-System Cooperation in document Annotation based on Information Extraction. In proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management, Siguenza, Spain, 65-80. (2002)
20. Yihong Ding, David W. Embley.: Using Data-Extraction Ontologies to Foster Automating Semantic Annotation. In proceedings of the 22nd International Conference on Data Engineering Workshops. (2006)
21. Liu Yuan, Zhanhuai Li, Shiliang Chen.: Ontology-Based Annotation for Deep Web Data. Journal of software, Vol. 19, No. 2, 237-245. (2008)
22. Yoo Jung An, James Geller, Yi-Ta Wu, Soon Ae Chun.: Semantic deep web: automatic attribute extraction from the deep web data sources. Symposium on Applied Computing, 1667-1672. (2007)
23. Chen Kerui, Zuo Wanli, Zhang Fan, He Fengling, Peng Tao.: Automatic generation of domain-specific ontology from deep web. Journal of Information and Computational Science, Vol. 7, No. 2, 519-525. (2010)
24. Yanghong Zhai, Bing Liu.: Web Data Extraction Based on Partial Tree Alignment. In proceedings of World Wide Web Conference 2005, 76-85. (2005)
25. Kerui Chen, Wanli Zuo, Fan Zhang, Fengling He, Yongheng Chen.: Robust and Efficient Annotation based on Ontology Evolution for Deep Web Data. Journal of Computers, unpublished. (2011)
26. Adwait Ratnaparkhi.: A maximum entropy model for part-of-speech tagging. In proceedings of the 1st Empirical Methods in Natural Language Processing Conference, 133–141. (1996)

27. Ariel Felner, Sarit Kraus Richard E. Korf.: KBFS: K-best-first search. Annals of Mathematics and Artificial Intelligence, Vol. 39, No. 1-2, 19-39. (2003)
28. Giorgos Flouris, Dimitris Plexousakis.: Bridging Ontology Evolution and Belief Change. Lecture Notes in Computer Science, Vol. 3955, 486-489. (2006)
29. W. Yang.: Identifying syntactic differences between two programs. Software Practice and Experience, Vol. 21, No. 7, 739-755. (1991)
30. WEIFENG SU, JIYING WANG, FREDERICK H. LOCHOVSKY.: ODE: Ontology-Assisted Data Extraction. ACM Transactions on Database Systems, Vol. 34, No. 2, 12-35. (2009)

Kerui Chen was born in 1983. She is a Ph.D. in the college of Computer Science and Technology at Jilin University. She current research interests include Web Intelligence, Ontology Engineering and Information integration.

Wanli Zuo was born in 1957. He is a professor and doctoral supervisor at Department of Computer Science and technology, Jilin University. Main research area covers Database Theory, Machine Learning, Data Mining and Web Mining, Web Search Engines, Web Intelligence. He was as a senior visiting scholar, engaged in collaborative research in Louisiana State University (USA) from 1996-1997. He was principle responsible member of 3 national NSFC programs. More than 130 papers of him were published in key Chinese journals or international conferences, 65 of which are cited by SCI/EI. He has five books published by the Higher Education Press of China. And he obtained 5 national and departmental awards. He is a member of System Software Committee of China's Computer Federation, prominent young and middle-aged specialist of Jilin Province.

Fengling He was born in 1962. He is a professor at the Jilin University and a CCF senior member. His research areas are database, data mining and Web search engine.

Yongheng Chen was born in 1979. He is a Ph.D. candidate in the college of Computer Science and Technology at Jilin University. He current research interests include Multi-Core, Database.

Ying Wang was born in 1981. She is a lecturer at the Jilin University and a CCF member. She received her Ph.D. degree from Jilin University. Her research area is Web Information Mining, Ontology and Web search engine.

Received: October 11, 2010; Accepted: January 19, 2011.

DOI: 10.2298/CSIS101019024Z

Extracting Minimal Unsatisfiable Subformulas in Satisfiability Modulo Theories

Jianmin Zhang, Shengyu Shen, Jun Zhang, Weixia Xu, and Sikun Li

Dept. of Computer Science, National University of Defense Technology
410073 Changsha, China
{jmzhang,syshen,junzhang,wxxu,skli}@nudt.edu.cn

Abstract. Explaining the causes of infeasibility of formulas has practical applications in various fields, such as formal verification and electronic design automation. A minimal unsatisfiable subformula provides a succinct explanation of infeasibility and is valuable for applications. The problem of deriving minimal unsatisfiable cores from Boolean formulas has been addressed rather frequently in recent years. However little attention has been concentrated on extraction of unsatisfiable subformulas in Satisfiability Modulo Theories(SMT). In this paper, we propose a depth-first-search algorithm and a breadth-first-search algorithm to compute minimal unsatisfiable cores in SMT, adopting different searching strategy. We report and analyze experimental results obtaining from a very extensive test on SMT-LIB benchmarks.

Keywords: Satisfiability Modulo Theories, minimal unsatisfiable subformula, depth-first-search, breadth-first-search.

1. Introduction

Formal verification has been based on efficient reasoning engines, such as Binary Decision Diagrams(BDD), and more recently propositional satisfiability(SAT) procedures, reasoning at Boolean level. Especially, during the last decade of impressive advances in the efficiency of SAT solvers, SAT-based method are now a fundamental technique in many industrial applications, including many steps of design flows for VLSI chips, such as equivalence checking, property verification, Auto Test Pattern Generation and static timing analysis etc. However the source of hardware design and verification problems has increasingly moved from Boolean level to higher levels: most designers work at register transfer level or even higher levels. The formalism of plain propositional logic is often not suitable or expressive enough for representing many real-world problems, including the verification of RTL or behavioral designs. The high-level structural information is identified as a suitable representation formalism for practical problems of many applications, and thus such problems are more naturally expressible as satisfiability problems in first-order theories, namely Satisfiability Modulo Theories (SMT).

Many real-life problems, arising in formal verification, electronic design, artificial intelligence and other areas, can be formulated as constraint satisfaction

problems, which can be translated into propositional or first-order formulas in conjunctive normal form(CNF). Modern SAT or SMT solvers are able to determine whether a large formula is satisfiable or not. When a formula is unsatisfiable, we are generally interested in a minimal explanation of infeasibility that excludes irrelevant information. Thus it is often required to find a minimal unsatisfiable subformula, or called a minimal unsatisfiable core, that is, an unsatisfiable subset if it becomes satisfiable whenever any of its clauses is removed. Localizing a minimal unsatisfiable subformula is necessary to determine the underlying reasons for the failure, and is used in many practical applications, including model checking on predicate abstraction[1], vacancy detection[2], error localization[3], and synthesizing circuits[4], etc.

In the past decade, there have been considerable research works in finding Boolean unsatisfiable cores[5–29]. However propositional logic is often not expressive enough for representing many practical problems, which can be more naturally addressed in the framework of SMT. The impressive advances of the computational power of SMT solvers makes it possible to extract the unsatisfiable cores from the formulas in SMT. Consequently, the development of effective methods for computing unsatisfiable subformulas in SMT is highlighted as an important goal for the research community. Although some SMT solvers support unsatisfiable subformulas generation, such as CVCLite[30], MathSAT[31] and Yices[32]. A simple and flexible algorithm [33] is the first published works on deriving unsatisfiable cores from formulas in SMT. However, as they said, one limitation of all these approaches is that the resulting unsatisfiable core is not guaranteed to be minimal.

In this paper, we tackle the problem of extracting minimal unsatisfiable cores from practical problem instances in SMT, by the depth-first-search algorithm (Depth-first-search Minimal Unsatisfiable Subformulas Extractor, DFS-MUSE) and the breadth-first-search algorithm (Breadth-first-search Minimal Unsatisfiable Subformulas Extractor, BFS-MUSE). To the best of our knowledge, they are the first works aiming at finding minimal unsatisfiable cores in SMT. Firstly, we present the definitions of the searching graph, and the live node, the dead node and the pending node in the searching graph. Next our algorithms construct the subformulas of original instance as a searching graph, and then recursively remove the clauses not included by the minimal unsatisfiable core from the original formula, adopting depth-first-search or breadth-first-search strategy. Simultaneously they change dynamically the order of variables in the subformula. Some pruning techniques are integrated into the algorithms to remove those unnecessary satisfiability checks as soon as possible, such as conflict clauses sharing and subformulas caching. To evaluate the efficiency of DFS-MUSE and BFS-MUSE, we have performed many experimental tests, and compared them on a large number of SMT-LIB problem instances coming from SMT solver competition benchmarks of CAV 2009.

The binaries of DFS-MUSE and BFS-MUSE are available for downloaded at http://www.ssypub.org/∼zjm/.

The paper is organized as follows. The next section surveys the related work on computing unsatisfiable cores. Section 3 introduces the basic definitions used throughout the paper. Section 4 gives the theoretical analysis for our algorithms. Section 5 proposes DFS-MUSE adopting depth-first-search strategy. Section 6 presents BFS-MUSE using breadth-first-search algorithm. Section 7 shows and analyzes experimental results on the benchmarks used by SMT solver competition of CAV 2009. Finally, Section 8 concludes the paper and outlines future research work.

2. Related Work

There have been many different contributions to research on unsatisfiable subformulas extraction in the last few years, owing to the increasing importance in numerous practical applications. An algorithm[5, 6] for deriving small unsatisfiable cores are based on the ability of DPLL-based SAT solvers to produce resolution refutations. In [7, 8], a method of adaptive core search guided by clauses hardness is employed to extract small unsatisfiable subformulas, but in [9] an exact minimal unsatisfiable core is obtained.

In [10], an algorithm of enumerating all possible subsets is suggested to compute a minimum unsatisfiable core. Another approach is called AMUSE[11], in which selector variables are added to each clause and the unsatisfiable core is derived by a branch-and-bound algorithm on the updated formula. A different algorithm that guarantees minimality is MUP[12], which is mainly a prover of minimal unsatisfiability, as opposed to an unsatisfiable core extractor. Some research works[13–18], based on a relationship between maximal satisfiability and minimal unsatisfiability, have developed some sound techniques for finding all minimal unsatisfiable cores or a minimum unsatisfiable core.

CoreTimmer[19, 20] iterates over each internal node that consumes a large number of clauses and attempts to prove them without these clauses. A scalable algorithm[21], adopting a deeper exploration of resolution refutation, is proposed for minimal unsatisfiable cores extraction. In [22–26], the authors present the algorithms which tracks minimal unsatisfiable subformulas according to the trace of a failed local search run for consistency checking.

A novel algorithm[27] to find minimal unsatisfiable cores is based on Brouwer's fixed point approximation theorem to satisfiability. In [28], the authors present a new framework called constructive, which is based on a combination of a local search procedure and an exhaustive DPLL-like algorithm. Two new resolution-based algorithms are proposed in [29]. The algorithms is used to compute a minimal unsatisfiable core or, if time-out encountered, a small non-minimal unsatisfiable core. These algorithms can be applied to either standard clause-level unsatisfiable subformulas extraction or high-level unsatisfiable subformulas extraction, that is, an extraction of an unsatisfiable subformulas in term of propositional constraints supplied by user application.

However, existing works have very little concern in the literature regarding extraction of unsatisfiable subformulas in SMT. In 2007, Cimatti et al[33] firstly

proposed an algorithm to compute the small unsatisfiable subformulas in SMT. It combines a SMT solver and an external propositional core extractor: Firstly, the SMT solver produces the theory lemmas found during the search; Secondly, the propositional core extractor is called on the boolean abstraction of the original SMT problem and of the theory lemmas, and then obtain a small unsatisfiable core of the original SMT formula. Just as the authors said, one limitation of this algorithm is that the resulting unsatisfiable core in SMT is not guaranteed to be minimal, even if external propositional core extractor returns minimal boolean unsatisfiable subformulas. Whereas up till now, there is no published work devoted to the minimal unsatisfiable subformulas extraction from problem instances in SMT.

3. Preliminaries

Satisfiability Modulo Theories arises in many industrial applications, especially in formal verification and electronic design automation. SMT is defined as the problem of determining the satisfiability of a quantifier-free first-order logic formula with respect to one or more of decidable theories. In a number of practical applications, SMT problem instances typically consist of logical combinations of atoms from different theories, such as the theory of integer linear arithmetic, the theory of arrays, the theory of equality with uninterpreted functions, set theory and so on.

In recent years, a large number of researchers are actively exploring a variety of approaches for solving SMT problems. Most of these methods are classified as eager or lazy techniques. In eager techniques, the input formula is translated using a satisfiability-preserving transformation into a propositional formula which is then checked by a SAT solver for satisfiability. The lazy techniques instead abstract each atom of the input formula by a distinct propositional variable, use a SAT solver to find a propositional model of the formula, and then check that model against the theory solvers. In a word, the substantial advances in algorithms and implementations of SMT solver for several years have inspired the quest of efficient solutions for the problem of minimal unsatisfiable subformulas extraction.

Next some basic definitions and notations used throughout the paper are given as follows:

Definition 1. ***(Unsatisfiable Subformula/Unsatisfiable Core).*** *Given an unsatisfiable formula φ, ψ is an unsatisfiable subformula(core) of φ iff ψ is an unsatisfiable formula and $\psi \subseteq \varphi$.*

Observe that an unsatisfiable core can be defined as any subset, which is infeasible, of the original formula. Consequently, there may exist many different unsatisfiable cores, with different number of clauses, for the same problem instance, such that some of these cores are subsets of others.

Definition 2. ***(Minimal Unsatisfiable Subformula/Minimal Unsatisfiable Core).*** *Given an unsatisfiable formula φ, and ψ is an unsatisfiable subformula(core) of*

φ. Then ψ is a minimal unsatisfiable subfromula(core) iff removing any clause $\omega \in \psi$ from ψ implies that $\psi - \{\omega\}$ is satisfiable.

According to the definition, an unsatisfiable subformula is minimal if it becomes satisfiable whenever any of its clauses is removed, in other words, all of its proper subsets are satisfiable. We should note that in many cases the size of an unsatisfiable subformula may be much larger than the size of a minimal unsatisfiable subformula, because the non-minimal unsatisfiable subformulas probably contain many redundant clauses which cannot be found by simple resolution rules.

4. Theoretical Analysis

We propose two algorithms to extract minimal unsatisfiable cores from the formulas in SMT. The searching strategy of two algorithms are respectively depth-first and breadth-fist. An efficient SMT decision procedure, which is based on the DPLL(*T*) decision scheme[34], is integrated into the algorithms of deriving unsatisfiable subformulas. Our algorithms thoroughly utilizes the information generated by the SMT solving process. The algorithms construct the subformulas of original instance in SMT as a searching graph, and derive the minimal unsatisfiable cores adopting different search ways. Firstly, the definition of the searching graph of a formula in SMT is given as follows:

Definition 3. ***(Searching graph).*** *Given an unsatisfiable formula φ in SMT, if a directed acyclic graph $G(V, E, s)$ satisfies the following conditions: (a) it contains only one sink node, which is on behalf of φ; (b) $\forall p \in V \setminus \{s\}$, the node p represents the formula $\psi = \wedge_1^n C_i$; If v is the k-th child node of p, the node v denotes the formula $\phi_k = \wedge_1^n C_i \setminus C_k$, where $v \in V$, $1 \leq k \leq n$; e_{pv} is an edge from the parent node p to the child node v, where $e_{pv} \in E$. Then $G(V, E, s)$ is called a searching graph of φ.*

Suppose φ is an unsatisfiable formula in SMT, $G(V, E, s)$ is the searching graph of φ. Furthermore, we can classify all of the nodes of $G(V, E, s)$ into three categories: the live nodes, the dead nodes and the pending nodes. The following shows the definitions of the three types of nodes.

Definition 4. ***(live node).*** *Given an unsatisfiable formula φ in SMT, and $G(V, E, s)$ is the searching graph of φ. Suppose $v \in V$, and ϕ denoted by v, where $\phi \subseteq \varphi$. Then v is a live node iff ϕ is unsatisfiable.*

Definition 5. ***(dead node).*** *Given an unsatisfiable formula φ in SMT, and $G(V, E, s)$ is the searching graph of φ. Suppose $v \in V$, and ϕ represented by v, where $\phi \subseteq \varphi$. Then v is a dead node iff ϕ is satisfiable.*

Definition 6. ***(pending node).*** *Given an unsatisfiable formula φ in SMT, and $G(V, E, s)$ is the searching graph of φ. Suppose $v \in V$. If the search process has not reach the node v, v is called a pending node.*

Next the definition of transition relation of the live nodes, the dead nodes and the pending nodes is given as follows:

Definition 7. ***(the transition of nodes).*** *Given an unsatisfiable formula φ in SMT, and $G(V, E, s)$ is the searching graph of φ. In $G(V, E, s)$, the transitions from pending nodes to dead nodes and live nodes are defined as: (a) pending nodes $\rightarrow$ dead nodes: When a subformula corresponding to a pending node is proved to be satisfiable, the pending node is changed to a dead node; (b) pending nodes $\rightarrow$ live nodes: When a subformula denoted by a pending node is unsatisfiable, the pending node is changed to a live node.*

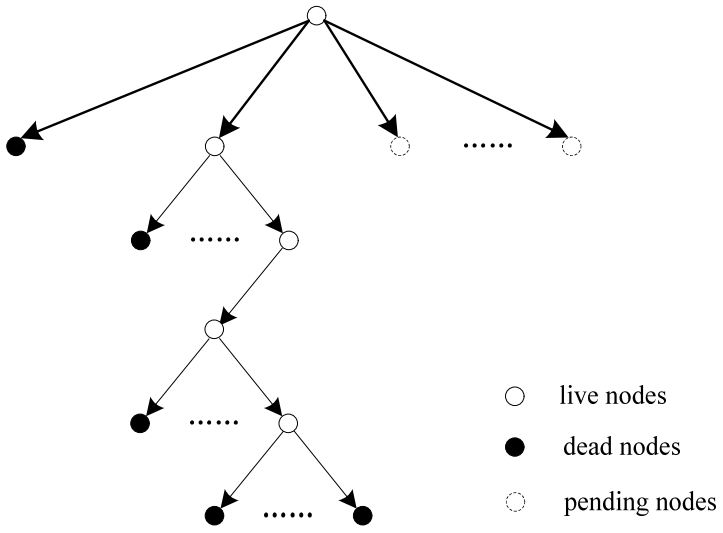


Fig. 1. An example of a searching graph including three types of nodes

Fig.1 shows the sketch of a searching graph including three types of nodes. Suppose φ is an unsatisfiable formula in SMT, and then DFS-MUSE or BFS-MUSE builds a searching graph $G(V, E, s)$, in which the original formula φ is represented by the sink node *s*, and each internal node corresponds to one of the subformulas of φ. However, what is the relationship between the minimal unsatisfiable cores of a formula and the nodes of the corresponding searching graph? According to the above definitions, we may come to the following conclusions.

Theorem 1. *Given an unsatisfiable formula φ in SMT, and $G(V, E, s)$ is the searching graph of φ. Then the subgraph, in which the sink corresponds to a dead node, cannot contain an unsatisfiable subformula.*

Proof. Proof by contradiction. Given a dead node $v \in V$, and ψ is the subformula corresponding to v, where $\psi \subseteq \varphi$. According to Definition 5, ψ is satisfiable. We assume there exists an unsatisfiable subformula $\phi \subseteq \psi$. Further n clauses $\{\omega_1, \cdots, \omega_n\}$ are joined into ϕ, and then we get η=$\phi \cup\{\omega_1, \cdots, \omega_n\}$, where $\{\omega_1, \cdots, \omega_n\} \subseteq \{\psi - \phi\}$. Since η contains all clauses belonging to ϕ, namely $\phi \subseteq \eta$, the truth assignment to the literals in η will eventually lead to a conflict. It indicates that η is unsatisfiable. Eventually, when $\{\omega_1, \cdots, \omega_n\}$=$\{\psi - \phi\}$, η= $\phi \cup \{\psi - \phi\}$=$\psi$ is unsatisfiable. However, v is a dead node and ψ is satisfiable. Therefore, it results in a contradiction, and the assumption is false.

Theorem 2. *Given an unsatisfiable formula φ in SMT, and $G(V, E, s)$ is the searching graph of φ. Then a subformula ϕ denoted by the live node v is a minimal unsatisfiable core, iff all children of v are the dead nodes, where $\phi \subseteq \varphi$ and $v \in V$.*

Proof. Given a live node $v \in V$, and ψ is the subformula corresponding to v, where $\psi = \wedge_1^n C_i$ and $\psi \subseteq \varphi$. According to Definition 4, ψ is an unsatisfiable core of φ. Firstly, suppose ϕ is a minimal unsatisfiable core of φ, and we try to prove all children of v are the dead nodes. According to Definition 2, $\phi_i = \psi - \{C_i\}$ is satisfiable, where $\forall 1 \leq i \leq n$, $C_i \in \psi$. Moreover from Definition 3, the i-th child node of v is represented by the subformula ϕ_i. Therefore, all children of v are the dead nodes. Secondly, we assume all children of v are the dead nodes, and try to prove ϕ denoted by the live node v is a minimal unsatisfiable core. According to Definition 3 and 5, $\forall 1 \leq k \leq n$, each $\phi_k = \wedge_1^n C_i - \{C_k\}$, corresponding to the k-th child node of v, is satisfiable. Eventually from Definition 2, ϕ is a minimal unsatisfiable core of φ.

Suppose φ is an unsatisfiable formula in SMT, our algorithms construct a searching graph $G(V, E, s)$ for φ, where the original formula φ is represented by the sink node, and each internal node corresponds to a subformula of φ. According to the above theorems, the algorithms recursively remove the clauses not included by the minimal unsatisfiable core from the original formula φ, respectively adopting depth-first-search or breadth-first-search strategy, until a live node with all children being dead nodes is found. Simultaneously they change dynamically the order of variables in the subformula. Some pruning techniques are integrated into the algorithms to remove those unnecessary satisfiability checks as soon as possible, such as conflict clauses sharing and subformulas caching.

5. Depth-first-search Algorithm

The depth-first-search algorithm, also called DFS-MUSE to compute the minimal unsatisfiable subformulas introduces the searching graph as an organizing framework. Based on the theorems, DFS-MUSE firstly builds a searching graph for the input formula in SMT. Then adopting the depth-first-search strategy, the algorithm heuristically remove the clauses not included by the minimal

unsatisfiable core from the original formula. During the search, a SMT decision procedure using lazy techniques is called to determine the satisfiability of subformulas. Simultaneously some techniques are employed to accelerate the searching process that the dead nodes are cached and the conflict clauses are shared among different subformuals. Moreover, the algorithm changes dynamically the order of variables in the subformula. The depth-first-search algorithm iterates over and over, and will not stop, until a living node with all children being dead nodes is found. Then the subformula denoted by this living node is a minimal unsatisfiable core. Fig.2 provides the pseudo code of DFS-MUSE.

```
DFS_MUSE(formula)
1    SmallUS = ComputeUS(formula)
2    if (SmallUS == formula) then
3        return formula
4    else
5        IsMinUS = VerifyMinimalUS(SmallUS)
6        if (IsMinUS) then
7            return SmallUS
8        else
9            MinimalUS = DFS_MUSE(SmallUS)
10           return MinimalUS

ComputeUS(formula)
1    ite = EliminateITE(formula)
2    abs = AbstratExpression(ite)
3    for (arity = 0; arity < formula.size; arity++) do
4        interim = abs
5        for (count = formula.size; count > 0; count--) do
6            SmallUS = GraphPruning(interim)
7            cnf = BooleanConversion(SmallUS)
8            IsSAT = SATSolve(cnf)
9            if (!IsSAT) then
10               return SmallUS
11       abs = DynamicVarOrder(abs)
12   return formula
```

Fig. 2. Pseudo code of the depth-first-search algorithm

The depth-first-search algorithm employs an incremental way: Firstly, it computes an unsatisfiable subformula; Further, it derives the minimal unsatisfiable subformula. The function called $ComputeUS$ returns an unsatisfiable core of the input formula. After getting an unsatisfiable core, DFS-MUSE judges and branches. If the returned unsatisfiable core is the input subformula named $formula$, $formula$ is the derived minimal unsatisfiable core. Otherwise, according to the above conclusion, the function called $VerifyMinimalUS$ is used to determine whether the unsatisfiable core $SmallUS$ is minimal or not. If $SmallUS$

is the derived minimal unsatisfiable core, DFS-MUSE will stop; or else the approach regards $SmallUS$ as the new input formula, and recursively computes the minimal unsatisfiable core in the depth-first-search way. When the order of branches changes, the depth-first-search algorithm may obtain different minimal unsatisfiable subformulas.

Fig.2 also shows the process of $ComputeUS$ to extract an unsatisfiable core from input formula in SMT. This function firstly builds a searching graph for the input formula, and then finds a live node in the depth-first-search way. The function called *EliminateITE* is to remove the ITE(If-Then-Else) terms from the formulas. Next *AbstratExpression* replaces the literals in the formula by the abstract variables. Then an unsatisfiable subformula is explored in the space of the searching graph. The function named *GraphPruning* is used to prune the redundant subformulas and clauses from the subgraph, by the way of sharing the conflict clauses and caching dead nodes to avoid the unnecessary satisfiability checks. *BooleanConversion* converts the formula to a Boolean formula, and a SAT solver with DPLL procedure engages in determining satisfiability of the Boolean formula. If unsatisfiable, we get an unsatisfiable subformula denoted by that live node. Otherwise the current node is dead and should be abandon. The function called *DynamicVarOrder* changes the order of variables in current subformula according to the frequency of false assignment. Finally, if the iteration is finished, the function will return the input formula itself as the derived unsatisfiable subformula. In Fig.2, *EliminateITE*, *AbstratExpression* and *BooleanConversion* are the functions belonging to a SMT decision procedure based on the DPLL(*T*) techniques.

6. Breadth-first-search Algorithm

The principles of the breadth-first-search algorithm are similar to the depth-first-search algorithm. BFS-MUSE firstly builds the searching graph for the original formula, and then explores a live node, of which all children are dead nodes, in the breadth-first way. The main difference between BFS-MUSE and DFS-MUSE is the searching strategy. BFS-MUSE moves horizontally on the branches of the graph. All of the same size of subformulas are firstly evaluated, and then the smaller ones are considered. DFS-MUSE instead decides the satisfiability of all subformulas with the size decreasing in the same subgraph at first, and then moves to the neighborhood subgraph. The pseudo code of BFS-MUSE, detailed in the later, is given in Fig.3.

The input of the breadth-first-search algorithm is a formula in SMT. The approach constructs a searching graph $G(V, E, s)$ for the original formula, and tries to find a live node with all children being dead nodes, adopting the breadth-first-search strategy. Some functions of the breadth-first-search algorithm, such as *EliminateITE* and *AbstratExpression*, are almost as same as the depth-first-search algorithm. The recursive function called *ComputeMinUS* is employed to extract a minimal unsatisfiable core from the input formula in SMT. There are some differences on the *GraphPruning* function between BFS-MUSE and

```
BFS_MUSE(formula)
1    ite = EliminateITE(formula)
2    abs = AbstratExpression(ite)
3    MinimalUS = ComputeMinUS(abs)
4    return MinimalUS

ComputeMinUS(formula)
1    IsMinUS = true
2    for (arity = 0; arity < formula.size; arity++) do
3        interim = GraphPruning(formula, arity)
4        cnf = BooleanConversion(iterim)
6        IsSAT = SATSolve(cnf)
5        if (!IsSAT) then
7            IsMinUS = false
8            break
9    if (IsMinUS) then
10       return formula
11    else
12       reordered = DynamicVarOrder(interim)
13       MinUS = ComputeMinUS(reordered)
14       return MinUS
```

Fig. 3. Pseudo code of the breadth-first-search algorithm

DFS-MUSE. In Fig.3, a parameter named *arity* is added into the *GraphPruning* function, and indicates the clause from which the function begins to prune the searching graph.

Then the *ComputeMinUS* function converts the pruned subgraph to Boolean formulas, and calls a SAT solver to check its satisfiability. If the formula is unsatisfiable, a child of current live node is also a live node, that is to say, a smaller unsatisfiable core is detected. Therefore, the subformula denoted by the current live node is not a minimal unsatisfiable core. Then the function makes the smaller live node as the new input, and recursively searches a minimal unsatisfiable subformula. The function will not halt until a live node with all children being dead nodes is found. That live node is the derived minimal unsatisfiable core. Similar to the depth-first-search algorithm, the breadth-first-search algorithm also can derive different minimal unsatisfiable subformulas while changing the order of branching.

7. Experimental Results and Analysis

To experimentally evaluate the effectiveness of our algorithms, we have selected a large number of problem instances from the well-known benchmark family[35], which is used in SMT solvers competition affiliated with CAV conference. We compare the depth-first-search algorithm with the breadth-first-search algorithm on these benchmarks. The inputs of two algorithms are the formulas

in SMT-LIB format. The experiments were conducted on a 2.5 GHz Athlon*2 machine having 2 GB memory and running the Linux operating system.

The depth-first-search algorithm and the breadth-first-search algorithm to find minimal unsatisfiable subformulas are implemented in C++. Some functions of an open-source SMT solver called ArgoLib[36] are integrated in our algorithms. The Argolib currently can solve satisfiability of the formulas, which consist of logic combination of atoms from some theories such as the theory of integer and real linear arithmetic(LIA/LRA), the theory of integer and real differential logic(IDL/RDL). The runtime is in seconds, and the value of timeout was set to 1800 seconds.

The experimental results[1] of the depth-first-search algorithm and the breadth-first-search algorithm on 15 typical formulas are listed in Table 1. Table 1 shows the number of variables (vars) and the number of clauses (clas) for each problem instance. Table 1 also provides the runtime in seconds (DFS-MUSE time) of DFS-MUSE and the number of clauses (DFS-MUSE size) in the derived minimal unsatisfiable subformula. The last two columns present the runtime in seconds (BFS-MUSE time) and the size of the minimal unsatisfiable subformula(BFS-MUSE size) extracted by BFS-MUSE.

Table 1. Performance results on 15 typical problem instances

Problem Instances	**vars**	**clas**	**DFS-MUSE**		**BFS-MUSE**	
			time	size	time	size
bad_echos_ascend.base	58	259	5.18	11	**5.03**	11
sc_init_frame_gap.base	58	265	**5.11**	13	5.14	13
good_frame_update.induction	89	439	29.00	161	**28.74**	161
good_frame_update.base	89	467	72.81	311	**67.75**	311
windowreal-safe2-2	37	404	0.73	188	**0.68**	188
windowreal-safe-2	37	404	0.75	195	**0.68**	195
lpsat-goal-1	83	1345	**1.67**	17	1.69	17
lpsat-goal-2	142	2650	**12.30**	1283	17.16	1283
lpsat-goal-3	201	3955	**43.47**	2548	68.55	2548
windowreal-no_t_deadlock-15	219	2933	**176.96**	1351	179.53	1351
windowreal-no_t_deadlock-16	233	3128	**208.13**	1441	236.58	1441
windowreal-no_t_deadlock-17	247	3323	**293.38**	1531	303.32	1531
windowreal-no_t_deadlock-18	261	3519	**347.24**	1622	391.02	1622
windowreal-no_t_deadlock-19	275	3714	**463.78**	1712	497.97	1712
windowreal-no_t_deadlock-20	289	3909	**547.44**	1802	633.77	1802

Fig.4 shows the experimental results of DFS-MUSE as compared with BFS-MUSE on inf-bakery-mutex benchmarks, coming from SMT solver competition

[1] More detailed performance results of two algorithms are available for downloaded at http://www.ssypub.org/∼zjm/

benchmarks. The size of 20 problem instances of inf-bakery-mutex benchmarks ranges from 65 to 1053 clauses.

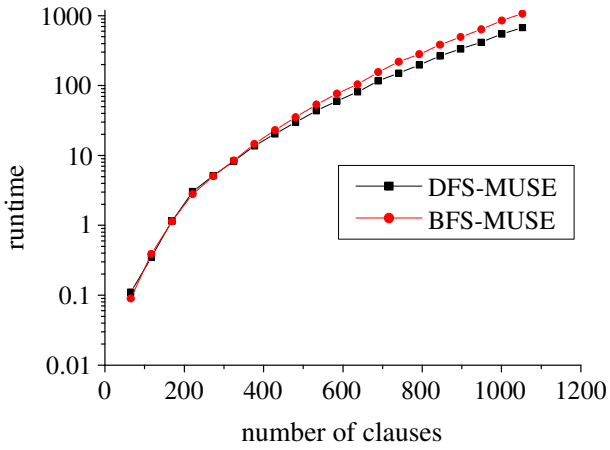


Fig. 4. Experimental results on inf-bakery-mutex benchmarks

Fig.5 shows the experimental results of the depth-first-search algorithm as compared with the breadth-first-search algorithm on windowreal-no_t_deadlock benchmarks, coming from SMT solver competition benchmarks. The size of 20 instances of windowreal-no_t_deadlock benchmarks ranges from 203 to 3908 clauses.

Fig.6 shows the experimental results of DFS-MUSE as compared with BFS-MUSE on pursuit-safety benchmarks, which comes from SMT solver competition benchmarks. The size of 16 instances of pursuit-safety benchmarks ranges from 113 to 1763 clauses.

Fig.7 shows the experimental results of the depth-first-search algorithm as compared with the breadth-first-search algorithm on gasburner-prop3 benchmarks, coming from SMT solver competition benchmarks. The size of 20 instances of gasburner-prop3 benchmarks ranges from 28 to 522 clauses.

From Fig.4 to Fig.7, there is a data point for a result of each formula computed by the depth-first-search algorithm and the breadth-first-search algorithm labelled with different icons, with the position along the vertical axis indicating the runtime in seconds of the algorithms, and the horizontal position indicating the number of clauses contained by every instance. Note that vertical axis has a logarithmic scale. The results of DFS-MUSE are denoted by box icon, and the results of BFS-MUSE are instead denoted by dot icon. The two curves re-

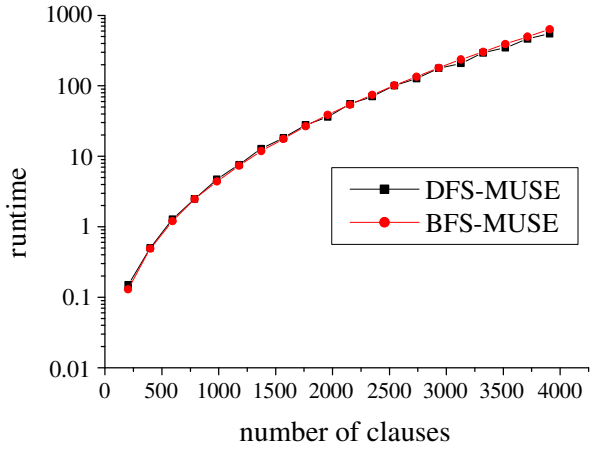


Fig. 5. Experimental results on windowreal-no_t_deadlock benchmarks

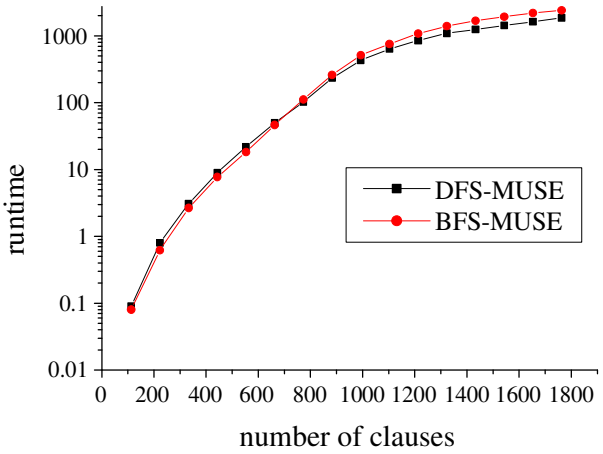


Fig. 6. Experimental results on pursuit-safety benchmarks

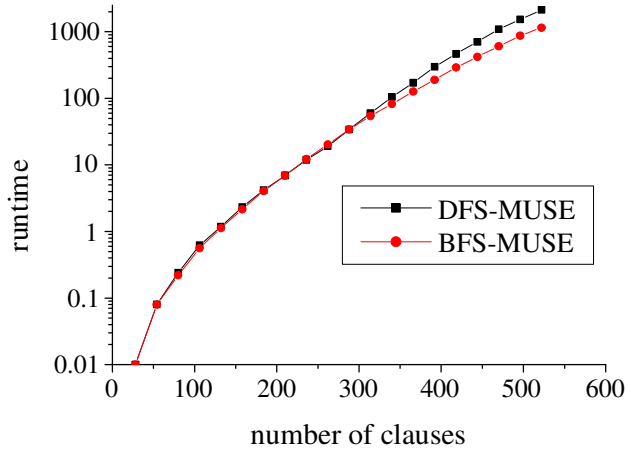


Fig. 7. Experimental results on gasburner-prop3 benchmarks

spectively are represented as the trends of runtime of two algorithms, with the increment of number of clauses included by the benchmarks.

Fig.8 gives the performance results of the depth-first-search algorithm and the breadth-first-search algorithm on more than 200 problem instances of SMT solver competition benchmarks. This figure is a log-log scatter plot that charts the runtime of DFS-MUSE along on the x-axis against the runtime of BFS-MUSE on the y-axis. In other words, each dot in this figure is denoted by the ratio of runtime of the depth-first-search algorithm and the breadth-first-search algorithm. The time unit of two axes is second, and the limit time of both tools was 1800 seconds.

From Table 1, we may observe the following. For all problem instances, the percentage of clauses in the minimal unsatisfiable subformulas is quite small, in most cases from 1% to 50%. Therefore, the minimal unsatisfiable subformulas can generally provide more succinct explanations of infeasibility, and is more valuable for a variety of practical applications.

From Table 1, we also can directly observe that the runtime of the depth-first-search algorithm and the breadth-first-search algorithm has only a little difference. From Fig.4 to Fig.7, we may come to the same conclusion, because two curves respectively corresponding to different approaches are close. Furthermore, the scatter dots are distributed closely around the diagonal in Fig.8. Then we may conclude that DFS-MUSE and BFS-MUSE effectively extract the minimal unsatisfiable cores from the formulas in SMT, and the performance of two approaches is at the same order of magnitude.

In Fig.4, Fig.5 and Fig.6, the breadth-first-search algorithm is generally outperforms the depth-first-search algorithm, when the number of clauses con-

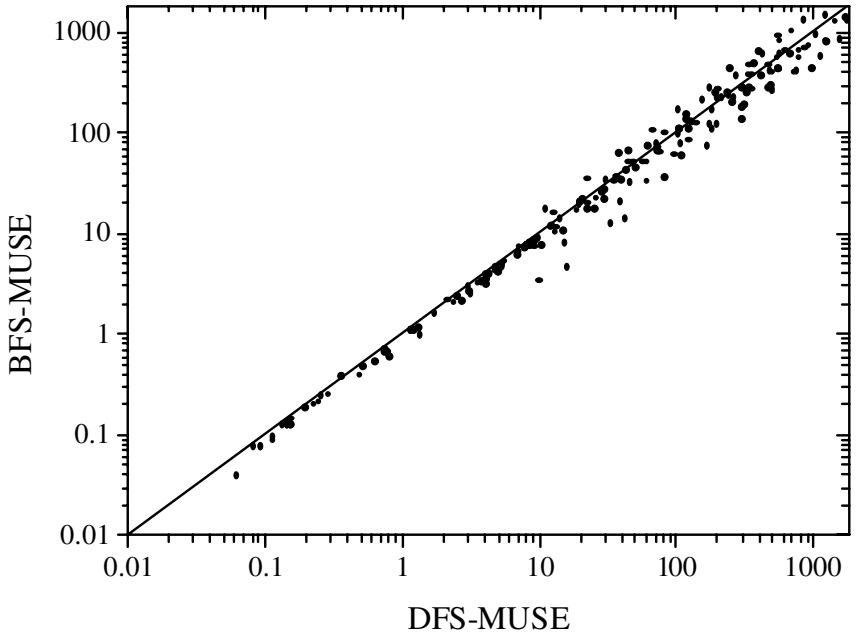


Fig. 8. Performance results on more than 200 instances

tained by original formula is small. However, with the size of formulas increasing, DFS-MUSE is more efficient, and the gap between DFS-MUSE and BFS-MUSE is enlarging. Moreover, in Fig.7, BFS-MUSE always outperforms DFS-MUSE since the size of all problem instances is comparatively small. In Fig.8, more and more scatter dots lie above the diagonal with the runtime increasing. Then we can reach a conclusion: When the size of the original formulas is small, BFS-MUSE is faster in most cases; But while the clauses in the formulas become more and more, DFS-MUSE is instead more efficient. The main causes are that BFS-MUSE is more simple for implementation and performs more moves per second, especially for those formulas with less clauses. On the other hand, after finding an unsatisfiable subformula, DFS-MUSE will continue to search the smaller ones along this subgraph, until reaching a minimal unsatisfiable subformula. This searching way of DFS-MUSE determines that it is more and more efficient while the problem instances containing more and more clauses.

8. Conclusion

We introduce the notation of a searching graph for a SMT formula, and three types of nodes in a searching graph: the live node, the dead node and the pending node. Using the searching graph as an organizing framework, we present

two algorithms called DFS-MUSE and BFS-MSUE to extract the minimal unsatisfiable cores from the formulas in SMT, respectively adopting depth-first-search and breadth-first-search strategy. DFS-MUSE and BFS-MUSE try to recursively remove the redundant clauses from the searching graph and derive a live node with all children being dead nodes. Some pruning techniques are integrated into the approaches, such as conflict clauses sharing and dead nodes caching.

A very extensive tests on SMT-LIB benchmarks are executed to evaluate the effectiveness and performance of DFS-MUSE and BFS-MUSE. The results show that the breadth-first-search algorithm generally outperforms the depth-first-search algorithm on smaller instances, and the depth-first-search algorithm is more efficient than the breadth-first-search algorithm while the formulas contain more and more clauses. The future works is to explore more aggressive techniques to prune the unnecessary satisfiability checks.

Acknowledgments. The authors would like to thank Filip Maric for his good suggestions and kind helps about using ArgoLib. The authors also thank all peer reviewers for their valuable comments and suggestions.This work is supported by the National Natural Science Foundation of China under grant No. 60603088.

References

1. Jain H, Kroening D.: Word level predicate abstraction and refinement for verifying RTL Verilog. In: Proceedings of the 42nd Design Automation Conference (DAC'05), pp. 445–450 (2005)
2. Simmonds J, et al.: Exploiting resolution proofs to speed up LTL vacuity detection for BMC. In: Proceedings of the 7th International Conference on Formal Methods in Computer Aided Design (FMCAD'07), pp. 3–12 (2007)
3. Stuckey PJ, Sulzmann M, Wazny J.: Improving Type Error Diagnosis. In: Proceedings of the 2004 ACM SIGPLAN workshop on Haskell, pp. 80–91 (2004)
4. Shen, S.Y., Qin, Y., Wang K.F., Xiao L.Q., Zhang J.M., Li, S.K.: Synthesizing complementary circuits automatically. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2010, 29(8): 1191–1202.
5. Zhang, L., Malik, S.: Extracting small unsatisfiable cores from unsatisfiable Boolean formula. In: Giunchiglia, E., Tacchella, A. (eds.) SAT 2003. LNCS, vol. 2919, Springer, Heidelberg (2004)
6. Zhang, L., Malik, S.: Validating SAT solvers using an independent resolution-based checker: practical implementations and other applications. In: Proceedings of 2003 Design, Automation and Test in Europe Conference (DATE2003), pp. 10880–10885 (2003)
7. Bruni, R., Sassano, A.: Restoring satisfiability or maintaining unsatisfiability by finding small unsatisfiable subformulae. Electronic Notes in Discrete Mathematics, 2001, 9: 162–173.
8. Bruni, R.: Approximating minimal unsatisfiable subformulae by means of adaptive core search. Discrete Applied Mathematics, 2003, 130(2): 85–100.
9. Bruni, R.: On exact selection of minimally unsatisfiable subformulae. Annals for Mathematics and Artificial Intelligence, 2005, 43: 35–50.
10. Lynce, I., Marques-Silva, J.P.: On computing minimum unsatisfiable cores. In: Hoos, H.H., Mitchell, D.G. (eds.) SAT 2004. LNCS, vol. 3542, pp. 305–310. Springer, Heidelberg (2005)

11. Oh, Y., Mneimneh, M.N., Andraus, Z.S., Sakallah, K.A., Markov, I.L.: AMUSE: a minimally-unsatisfiable subformula extractor. In: Proceedings of the 41st Design Automation Conference (DAC'04), pp. 518–523 (2004)
12. Huang, J.: MUP: A minimal unsatisfiability prover. In: Proceedings of the 10th Asia and South Pacific Design Automation Conference (ASPDAC'05), pp. 432–437 (2005)
13. Liffiton, M.H., Sakallah, K.A.: On finding all minimally unsatisfiable subformulas. In: Bacchus, F., Walsh, T. (eds.) SAT 2005. LNCS, vol. 3569, pp. 173–186. Springer, Heidelberg (2005)
14. Bailey, J., Stuckey, P.J.: Discovery of minimal unsatisfiable subsets of constraints using hitting set dualization. In: Hermenegildo, M.V., Cabeza, D. (eds.) PADL 2005. LNCS, vol. 3350, pp. 174–186. Springer, Heidelberg (2005)
15. Mneimneh, M.N., Lynce, I., Andraus, Z.S., Marques-Silva, J.P., Sakallah, K.A.: A branch and bound algorithm for extracting smallest minimal unsatisfiable formulas. In: Bacchus, F., Walsh, T. (eds.) SAT 2005. LNCS, vol. 3569, pp. 393–399. Springer, Heidelberg (2005)
16. Zhang, J.M., Li, S.K., Shen, S.Y.: Extracting minimum unsatisfiable cores with a greedy genetic algorithm. In:Sattar, A., Kang, B.H. (eds.) AI 2006. LNCS, vol. 4304, pp. 847–856. Springer, Heidelberg (2006)
17. Liffiton, M.H., Sakallah, K.A.: Algorithms for Computing Minimal Unsatisfiable Subsets of Constraints. Journal of Automated Reasoning. 2008, 40(1): 1–30.
18. Liffiton, M.H., Mneimneh, M.N., Lynce, I., Andraus, Z.S., Marques-Silva, J.P., Sakallah, K.A.: A branch and bound algorithm for extracting smallest minimal unsatisfiable formulas. Constraints, 2009, 14(4) 415–442.
19. Gershman, R., Koifman, M., Strichman, O.: Deriving small unsatisfiable cores with dominator. In: Ball, T., Jones, R.B. (eds.) CAV 2006. LNCS, vol. 4144, pp. 109–122. Springer, Heidelberg (2006)
20. Gershman, R., Koifman, M., Strichman, O.: An approach for extracting a small unsatisfiable core. Formal Methods in System Design, 2008, 33(1): 1–27.
21. Dershowitz, N., Hanna, Z., Nadel, A.: A scalable algorithm for minimal unsatisfiable core extraction. In: Biere, A., Gomes, C.P. (eds.) SAT 2006. LNCS, vol. 4121, pp. 36–41. Springer, Heidelberg (2006)
22. Gregoire, E., Mazuer, B., Piette, C.: Tracking MUSes and strict inconsistent covers. In: Proceedings of the the Sixth Conference on Formal Methods in Computer-Aided Design (FMCAD'06), pp. 39–46 (2006)
23. Gregoire, E., Mazuer, B., Piette, C.: Extracting MUSes. In: Proceedings of the 17th European Conference on Artificial Intelligence (EAI'06), pp. 387–391 (2006)
24. Gregoire, E., Mazuer, B., Piette, C.: Local-search extraction of MUSes. Constraints, 2007, 12(3): 325–344.
25. Gregoire, E., Mazuer, B., Piette, C.: Boosting a complete technique to find MSS and MUS thanks to a local search oracle. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence(IJCAI'07), pp. 2300–2305 (2007)
26. Gregoire, E., Mazuer, B., Piette, C.: Using local search to find MSSes and MUSes. European Journal of Operational Research, 2009, 199(3): 640–646.
27. Maaren H, Wieringa S.: Finding guaranteed MUSes fast. In: Buning HK, Zhao X, (eds.) SAT 2008. LNCS, vol. 4996, pp. 291–304. Springer, Heidelberg (2008)
28. Pitte, C., Hamadi, Y., Sais, L.: Efficient combination of decision procedures for MUS computation. In: Ghilardi, S., Sebastiani, R. (eds.) FroCos 2009, LNCS, Vol. 5749, pp. 335–349. Springer, Heidelberg (2009)

29. Nadel, A.: Boosting minimal unsatisfiable core extraction. In: Proceedings of the 10th International Conference on Formal Methods in Computer Aided Design (FM-CAD'10), pp. 221–229 (2010)
30. Barrett, C., Berezin, S.: CVC Lite: A New Implementation of the Cooperating Validity Checker. In: Alur, R., Peled, D.A. (eds.) CAV 2004, LNCS, Vol. 3114, pp. 515–518. Springer, Heidelberg (2004)
31. Bozzano M., Bruttomesso R., Cimatti A., Junttila T., van Rossum P., Schulz S., and Sebastiani R.: An incremental and Layered Procedure for the Satisfiability of Linear Arithmetic Logic. In: Halbwachs, N., Zuck, L. (eds.) TACAS 2005, LNCS, Vol. 3440, pp. 317–333. Springer, Heidelberg (2005)
32. Dutertre B., de Moura, L.: A Fast Linear-Arithmetic Solver for DPLL(T). In: Ball, T., Jones, R.B. (eds.) CAV 2006. LNCS, vol. 4144, pp. 81–94. Springer, Heidelberg (2006)
33. Cimatti A, Griggio A, Sebastiani R. A simple and flexible way of computing small unsatisfiable cores in SAT modulo theories. In: Marques-Silva J, Sakallah K, (eds.) SAT 2007. LNCS, vol. 4501, pp. 334–339. Springer, Heidelberg (2007)
34. Nieuwenhuis R, Oliveras A, Tinelli C. Solving SAT and SAT modulo theories: from an abstract Davis-Putnam-Logemann -Loveland procedure to DPLL(T). Journal of the ACM, 2006, 53(6): 937–977.
35. Barret C, Deters M, Oliveras A, et al.: http://www.smtcomp.org/2009/benchmarks.shtml, accessed in 2010.
36. Maric F, Janicic P.: Argo-lib: a generic platform for decision procedures. In: Basin, D., Rusinowitch, M. (eds.) IJCAR 2004, LNCS, vol. 3097, pp. 231–217. Springer, Heidelberg (2004)

Jianmin Zhang was born in China in 1979. He is a lecturer in computer science at National University of Defense Technology. He received Ph. D. degree in computer science from National University of Defense Technology, Changsha, China, in 2008. His major fields of study include constraint satisfaction and formal verification of hardware.

Shengyu Shen was born in China in 1975. He is a associate professor in computer science at National University of Defense Technology. He received Ph. D. degree in computer science from National University of Defense Technology, Changsha, China, in 2005. His major fields of study include formal verification of hardware.

Jun Zhang was born in China in 1975. He is Ph. D. candidate in computer science, at National University of Defense Technology, Changsha, China, from 2007. His major fields of study include formal verification of VLSI chips.

Weixia Xu was born in China in 1963. He is a professor in computer science at National University of Defense Technology. His major fields of study include computer architecture and VLSI verification.

Sikun Li was born in China in 1941. He is a professor in computer science at National University of Defense Technology. His major fields of study include VLSI designing and verification.

Received: October 19, 2010; Accepted: March 12, 2011

DOI: 10.2298/CSIS100407025J

Indexing Temporal Information for Web Pages

Peiquan Jin, Hong Chen, Xujian Zhao, Xiaowen Li, and Lihua Yue

School of Computer Science and Technology,
University of Science and Technology of China, 230027, Hefei, China
jpq@ustc.edu.cn

Abstract. Temporal information plays important roles in Web search, as Web pages intrinsically involve crawled time and most Web pages contain time keywords in their content. How to integrate temporal information in Web search engines has been a research focus in recent years, among which some key issues such as temporal-textual indexing and temporal information extraction have to be first studied. In this paper, we first present a framework of temporal-textual Web search engine. And then, we concentrate on designing a new hybrid index structure for temporal and textual information of Web pages. In particular, we propose to integrate B+-tree, inverted file and a typical temporal index called MAP21-Tree, to handle temporal-textual queries. We study five mechanisms to implement a hybrid index structure for temporal-textual queries, which use different ways to organize the inverted file, B+-tree and MAP-21 tree. After a theoretic analysis on the performance of those five index structures, we conduct experiments on both simulated and real data sets to make performance comparison. The experimental results show that among all the index schemes the first-inverted-file-then-MAP21-tree index structure has the best query performance and thus is an acceptable choice to be the temporal-textual index for future time-aware search engines.

Keywords: Web search, temporal-textual query, temporal information, index structure.

1. Introduction

Web search engines such as Google and Bing have been an important part in people's life. Most people rely on Google to find useful information. The major goal of search engine is to deliver right information to right users quickly, which is generally implemented by a query processing system. In order to achieve this goal, search engines provide many effective ways for users to express their queries precisely, and also develop some efficient algorithms in ranking and indexing. However, previous research on Web search does not pay enough attention to the temporal information in Web pages. For example, it is difficult to express queries like "to find the discount information about Nike in the next week" in Google. On the other side, time is one of essential characteristics of information [1], and most Web pages are related with

temporal information, e.g., business news, discount information and so on. Recently, time has been a focus in the area of Web information extraction [2]. Therefore, it is useful and meaningful to utilize temporal information in Web search to enhance traditional search engines, that is, to develop a temporal-textual Web search engine.

In this paper, we focus on the index structures for temporal-textual Web search. Our basic idea is to develop an efficient hybrid index structure to cope with temporal-textual queries, which makes an integration of traditional temporal index and textual index. The most famous textual index is the inverted file structure, so in this paper we use this structure as the basic textual index structure. For temporal index, we adopt the MAP21-Tree [3], which is an efficient temporal index structure in temporal database area. However, there are many choices when integrating inverted file with MAP21-Tree, and in some case we need to introduce B+-Tree as the index structure for update time. Hence, we aims at making a comparison study on those different integration mechanisms, and finally get the best hybrid index structure which has the best performance for temporal-textual queries.

A previous short version of this paper has been published in APWeb 2011 [32]. The major differences between this paper and the previous one are that in this paper we extend the design consideration of temporal-textual Web search engine as well as a time ontology for Web pages. Moreover, we conduct theoretical analysis and further experiments based on a synthetic and a real data set and use more metrics to make performance comparison on the index structures concerned. The main contributions of the paper can be summarized as follows:

(a) We classified the temporal information of Web pages into update time and content time, and introduced the new concept, primary time, into the index process;

(b) We studied and compared five hybrid index structures based on B+-tree, inverted file and MAP21-tree;

(c) We carried out large-scale experiments based on both simulation dataset and real dataset to evaluate the performance of our index structures.

The remainder of this paper is organized as follows. Section 2 introduces the framework of our temporal-textual Web search engine and its main components. Section 3 discusses the five hybrid index structures. Section 4 provides the experimental results. Section 5 describes related work. Finally, we conclude the paper and discuss our future work in Section 6.

2. A Temporal-Textual Web Search Engine

2.1. Time Ontology for Web Pages

Our temporal-textual Web search engine is based on a temporal ontology for Web, which supports different types of Web temporal information (as shown in Fig.1).

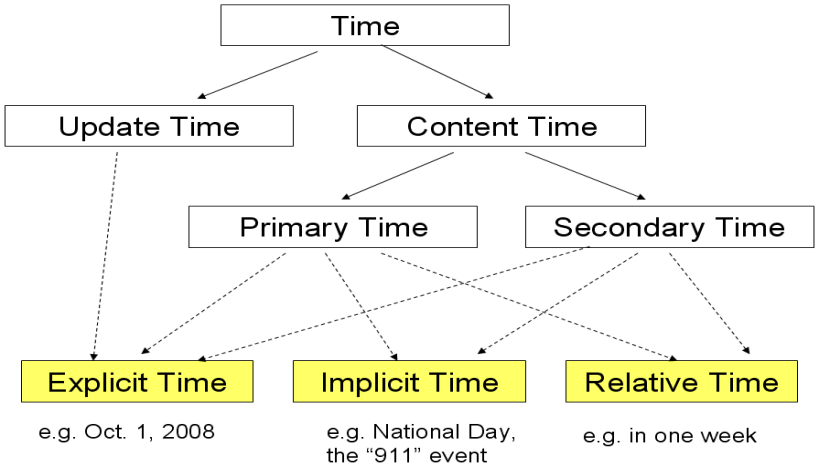


Fig. 1. The Time ontology for Web pages

(a) *Update time*: the update time of a Web page is defined as the crawled date of the Web page.

(b) *Content time*: the content time of a Web page is defined as the temporal information embedded in the main text of the Web page. We use a set of intervals to represent the content time.

(c) *Explicit time*: explicit time can directly be laid in the timeline. Basically, explicit time is a direct entry in the timeline and need not to be transformed.

(d) *Implicit time*: implicit time is a type of fuzzy time which can be mapped as an entry in the timeline with help of some predefined knowledge. Typical implicit time is holiday name or specific event. For example, the "911" event implies an implicit date, which is 2001/09/11.

(e) *Relative time*: relative time is one type of content time of a Web page. It must rely on another time of the Web page to resolve itself as an entry in the timeline. For example, the words "in three days" implies a relative time, since it must rely another date in the text to get the exact date.

(f) *Primary time*: primary time is one type of content time of a Web page. In detail, since there are several content times in one Web page, we will find the

most appropriate time that describes the events of the Web page from the set of content times. The most appropriate time is defined as primary time. Generally, when users search in Web through some temporal predicates, they want to get those pages whose primary time of contents is most close to the given temporal information.

(g) *Secondary time*: all the content time of a Web page can be regarded as secondary time, except the primary time.

These different types of time form a relatively complete framework for the representation of temporal information in Web pages. Among those types of time, the update time is always explicit time, since the crawled date of a Web page is definite. The content time can be explicit, implicit or relative. The content time usually contains a set of time period, one of which is chosen as the primary time and the others are secondary time. In our temporal-textual search engine, we only consider the update time and primary time. The reason is that when users search in Web through some temporal predicates, they usually want to get those pages whose primary time is most close to the given temporal information. Otherwise, the temporal predicates in queries will have little influence on the filtering of results, because the contents of most Web pages may contain a lot of and a large range of temporal information.

2.2. The Framework for Temporal-Textual Web Search Engine

Fig.2 shows the system architecture of the temporal-textual search engine. The search engine contains five modules.

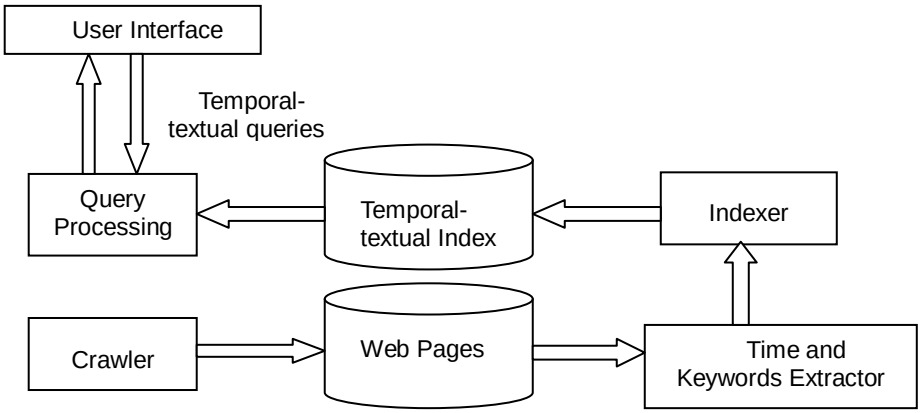


Fig. 2. The system architecture of the temporal-textual search engine

2.2.1 Web Crawler

The Web Crawler periodically crawl the Web to gather Web pages for further information extraction and retrieval. This module is similar with other spiders, except that it stores the crawled pages according to their crawled dates. For example, all the pages obtained in 2008/06/26 will be stored in the directory named “2008-06-26”. In the next extraction step, we will resolve the directory name to get the update time of a Web page.

2.2.2 Time and Keywords Extractor

In this module, the update time, content time, and the keywords of crawled Web pages will be extracted. For update time extraction, this module resolves the directory name and transforms it into standard date format. Since we use the crawled date as directory name, it is easy to get the update time of each Web page. For content time, we use temporal constraints to recognize content time and finally detect the primary time. We first analyze the Web page into a DOM tree. For each leaf node of the DOM tree, we use the TIMEX2 [13] to get the temporal information. After this procedure, we use some temporal constraints to detect the primary time among the whole set of content time. Temporal constraints are rules for primary time. For example, a temporal constraint may be “if a date appears in title, then this date is treated as primary time”. In this module, a set of keywords are also be extracted, which is based on traditional national language processing tools.

2.2.3 Index Constructor

When primary time and keywords are extracted from Web pages, the Index Constructor will create a hybrid temporal-textual index for those crawled Web pages. The hybrid index is based on three basic structures, namely B+-tree, inverted file, and MAP21-tree. We will discuss indexing techniques in detail in Section 4.

2.2.4 Query Processing

The Query Processing module processes user queries and returns ranked results. The main difference between this module and other search engines is that it supports temporal topological queries and uses new rank algorithm. It supports absolute temporal relation and relative temporal relation [30]. For example, it can process such as a user query as “find NBA news after 2008-06-30”, where an absolute temporal relation BEGIN(2008-06-30) is detected and processed. We also design a new ranking algorithm to sort the returned pages. The new ranking algorithm takes into account three factors: temporal information, text similarity, and page importance.

2.2.5 User Interface

The User Interface provides input ways for both temporal queries and textual queries. The textual queries are inputted by a single text box, which is similar with Google. The temporal queries are inputted by a time period box consisting of a start date and an end date. The returned results are shown based on a Timeline [31].

3. Indexer

The indexer is to build hybrid index structures to integrate text keywords and temporal information of Web pages. The inverted file is a standard technique for text indexing, so we adopt it as the basic index structure for text keywords in Web pages. The temporal information contains update time and primary time, in which the update time is regarded as a time instant and the primary time is modeled as a time period. The time granularity is set to day. As the update time is a time instant, we can use B+-tree to organize them or directly put them into inverted files. For the primary time, we adopt the MAP21-tree [3] as the basic index structure. MAP21-tree is designed towards time period and has better performance than other temporal indexes such as R-tree [22].

Table 1. The description of symbols

Symbol	Description
U	The number of update time in the time datasets
P	The number of primary time in the time datasets
K	The number of keywords in the lexicon
$P_U(u)$	The length of the page list of a update time u
$P_P(p)$	The length of the page list of a primary time p
$P_K(k)$	The length of the page list of a keyword k
B_{List}	Storage of page lists
$B_{Tree}(x)$	Storage of a tree of x elements
$T_{I/O}$	The time cost of disk accesses
T_{disk}	The time cost of one disk access
$T_{Tree}(x)$	The time cost to retrieve a tree of x elements
$T_{mg}(x)$	The time cost to merge x elements

We study five hybrid methods: (a) inverted file, B+-tree and MAP-21 triple index, (b) first inverted file then MAP21-tree and B+-tree, (c) first inverted file

then MAP21-tree, (d) expanded inverted file, (e) first MAP21-tree then inverted file. In additional, we emphasize the forth hybrid method expanded inverted file. Because the index structure of most related work about temporal text indexing is usually based on inverted file index, expanded inverted file can be considered as the similar method the previous work on building temporal index proposed. We will describe the hybrid index structures and present cost models for each structure. The symbols used in the cost models are listed in Table 1.

3.1. Inverted File, B+-tree and MAP21-tree Triple Index

In the first mechanism, we build indexes separately for keywords, update time and primary time, as shown in Fig.3. The keywords are indexed by an inverted file, which consists of a vocabulary, commonly organized as a B+-tree, and a posting list representing the information about each Web page. The update time is indexed by a B+-tree, while the primary time is organized as a MAP21-tree. Each leaf node of the three trees all points to their corresponding page lists.

A temporal-textual Web search comprises non-temporal keywords and temporal query types. Non-temporal query keywords are retrieved similar to conventional inverted files, temporal query types are passed to the B+-tree for update time and the MAP21-tree for primary time. The final results are produced by merging the page lists from three indexes.

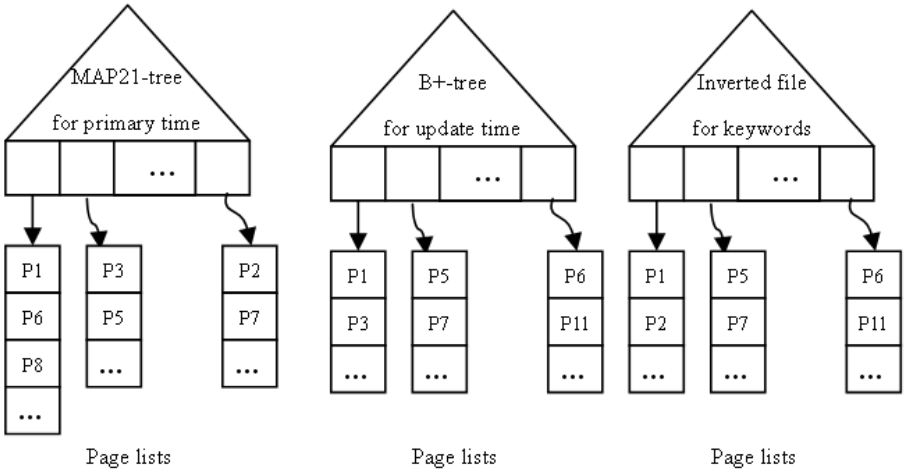


Fig. 3. Illustration of inverted file, B+-tree and MAP21-tree triple Index

The storage in disk comprises the three kinds of page lists and three trees, i.e. $Storage_1 = B^1_{Tree} + B^1_{List}$.

The storage of a tree that has x leaf nodes is $B_{Tree}(x) = O(x)$.

The storage of page lists depends on the length of each list, whose unit is the identifier of a page. Assuming the length of the list whose entry is keyword k is $P_K(k)$, the length of the list whose entry is update time u is $P_U(u)$ and the length of the list whose entry is primary time p is $P_P(p)$, the total length of all lists is $\sum_{u=1}^{U} P_U(u) + \sum_{p=1}^{P} P_P(p) + \sum_{k=1}^{K} P_K(k)$.

Then we have $B_{List}^1 = O\left(\sum_{u=1}^{U} P_U(u) + \sum_{p=1}^{P} P_P(p) + \sum_{k=1}^{K} P_K(k)\right)$.

So, $Storage_1 = B_{Tree}^1 + B_{List}^1$

$$= O(U+P+K) + O\left(\sum_{u=1}^{U} P_U(u) + \sum_{p=1}^{P} P_P(p) + \sum_{k=1}^{K} P_K(k)\right)$$

$$= O\left(\sum_{u=1}^{U} P_U(u) + \sum_{p=1}^{P} P_P(p) + \sum_{k=1}^{K} P_K(k)\right)$$

We can see the main cost of storage in disk is determined by page lists above and the levels of trees. The storage is mainly depended on the total length of all page lists.

Assuming there is a query including m keywords and temporal query types. The online computation includes: (a) the retrieval of the m page lists based on the *m* keywords, (b) the retrieval of the *n* page lists based on the given update time, (c) the retrieval of the *l* page lists based on the given primary time, and (d) the merge of the (*m* + *n* + *l*) page lists to return final results.

The time of loading page lists is determined by the number and total length of page lists. The merge processing is mainly the total length of these page lists.

For the tree that in this structure has U leaf nodes, assuming that the query time is $T_T(U)$.

The merge time for x elements in memory is $T_{mg} = O(x)$.

The time to read a page list containing x bytes is $T_{I/O} = T_{disk} \cdot O\left(x / B_{\text{section}}\right)$.

In the above equation, B_{section} is the page size, which depends on the file system. In our system it is 4K bytes.

Then, $Time_1 = T_{Tree}^1 + T_{I/O}^1 + T_{mg}^1$

$$= T_{Tree}(U+P+K) + \left(\sum_{i=1}^{m} T_{disk} \cdot O\left(P_U(u_i) / B_{\text{section}}\right)\right)$$
$$+ \left(\sum_{i=1}^{n} T_{disk} \cdot O\left(P_P(p_i) / B_{\text{section}}\right)\right) + \left(\sum_{i=1}^{l} T_{disk} \cdot O\left(P_K(k_i) / B_{\text{section}}\right)\right)$$
$$+ O\left(\sum_{i=1}^{m} P_U(u_i) + \sum_{i=1}^{n} P_P(p_i) + \sum_{i=1}^{l} P_K(k_i)\right)$$

For different queries, the query time is determined by three factors. The first factor is the searching time on the three index trees. The second is the merge operations of *m* page lists returned by the keywords query, *n* page lists returned by the update time query, and *l* page lists returned by the primary time query. The third factor is the time to read the (*m* + *n* + *l*) page lists from disk.

3.2. First Inverted File Then MAP21-tree and B+-tree

In this mechanism, two structures are maintained (as shown in Fig.4). The first one is an inverted file, each of whose leaf nodes points to a MAP21-tree which contains the primary time in the posting lists corresponding to the leaf node. The second one is a B+-tree used for indexing update time.

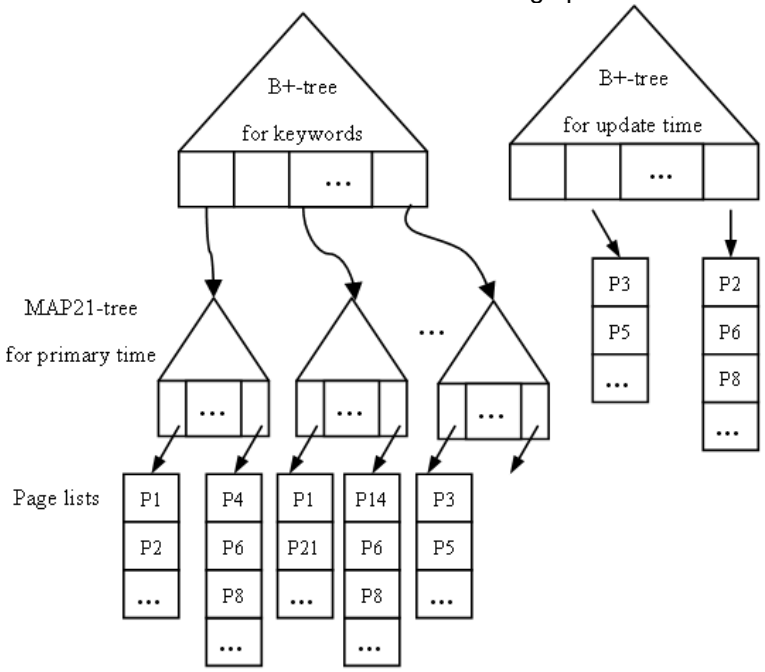


Fig. 4. Illustration of first inverted file then MAP21-tree and B+-tree index structure

Assuming $P_T(t)$ is the length of a page list whose entry is a ptime-keyword *p*.

The storage in disk includes these page lists, MAP21-trees pointed by K keywords and one B+-tree besides the page list whose entry is update time and one B+-tree. So,

$$Storage_2 = B^2_{Tree} + B^2_{List}$$
$$= K \cdot O(P) + O(U) + O(K) + O\left(\sum_{t=1}^{T} P_T(t) + \sum_{u=1}^{U} P_U(u)\right)$$

In fact, a MAP21-tree in this structure may not index all P leaves as in the first structure, so the scale of MAP21-trees is smaller. Thus we can see the cost of storage in disk is mainly caused by the total length of page lists whose entry is ptime-keyword and the length of page list whose entry is update time.

If we input *m* keywords, one primary time period and one update time instant, the online computation has three parts: (a) finding the *m* leaf nodes in the invert file according to the *m* query keywords, and then searching the corresponding MAP21-subtrees pointed by the *m* leaf nodes, and then loading the corresponding page lists from disk; (b) searching the B+-tree for the update time, and loading the corresponding page lists from disk; (c) merging the page lists returned by MAP21-tree and those returned by the update time B+-tree to generate final results. So,

$$Time_2 = T^2_{Tree} + T^2_{I/O} + T^2_{mg}$$
$$= m \cdot T_{Tree}(\overline{M}) + T_{Tree}(u) + \left(\sum_{i=1}^{m} T_{disk} \cdot O\left(P_T(u_i) / B_{section}\right)\right)$$
$$+ \left(\sum_{i=1}^{n} T_{disk} \cdot O\left(P_U(u_i) / B_{section}\right)\right) + O\left(\sum_{i=1}^{m} P_T(t_i) + \sum_{i=1}^{n} P_U(u_i)\right)$$

In the above equation, $\overline{M}$ is the average number of leaf nodes in all the *m* MAP21-trees.

There are three factors that affect the query time. The first is the retrieval of *m* MAP21-subtrees. The second is the merge operations of *m* page lists returned by *m* keywords. The third factor is the time to read the (*m* + *n*) page lists from disk, supposing the searching on the update time B+-tree returns *n* page lists.

3.3. First Inverted File Then MAP21-tree

Inverted file is built by B+-tree, the leaf node of B+-tree points to the inverted list of the keyword. In the structure, the leaf node points to a MAP21-tree which is build on time information of the inverted list of the corresponding keyword, then the leaf node of the MAP21-tree points to a set of page lists whose entry is depended on a pair of a keyword and a primary time. So n leaf nodes of B+-tree points totally to not n page lists but n MAP21-subtree. A pair of a keyword and a primary time period is named primary-keyword. We can know every MAP21-subtree in the structure may not index all primary times, because each word doesn't exist in every Web page, then it doesn't include all primary times, the MAP21-subtree is smaller than one unique MAP21-tree and the size of page lists pointed by leaf node of MAP21-subtree is almost smaller than MAP21-tree's. To update time, we don't index the type of time

information using B+-tree like the structure above, it is directly inserted the detail of a page list of MAP21-tree, which is the difference from the above index structure. We show the structure in Fig.5.

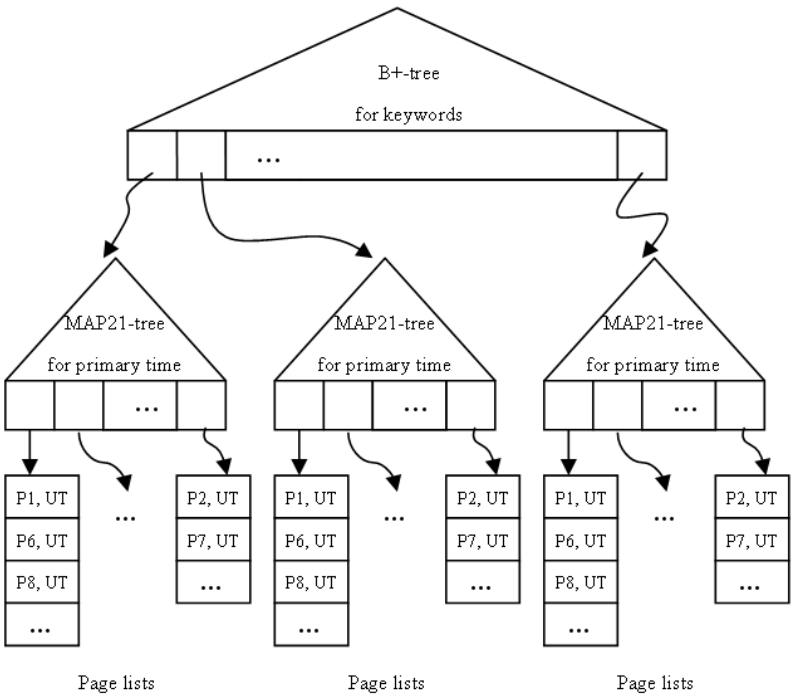


Fig. 5. The illustration of first inverted file then MAP21-tree(UT = Update Time)

The storage in disk includes these page lists, MAP21-trees pointed by K keywords and one B+-tree. So,

$$Storage_3 = B^3_{Tree} + B^3_{List} = K \cdot O(P) + O(U) + O\left(\sum_{t=1}^{T} P_T(t)\right)$$

The main cost of storage in disk is caused by the total length of page list whose entry is a putime-keyword.

If we input m keywords, one primary time period and a update time instant, the online computation has three parts: (a) to retrieve the m query keywords and then search the corresponding MAP21-subtrees by these leaf nodes of m keywords, and the loading of corresponding page lists m from the disk; (b) to retrieve the map value of the update time instant from corresponding page lists n which is in memory from disk; (c) to merge the n page lists. So,

$$Time_3 = T^3_{Tree} + T^3_{I/O} + T^3_{mg}$$

$$= m \cdot T_{Tree}\left(\overline{M}\right) + \left(\sum_{i=1}^{m} T_{disk} \cdot O\left(P_U\left(u_i\right) / B_{section}\right)\right) + O\left(\sum_{i=1}^{m} P_T\left(t_i\right)\right)$$

There are three factors that effect the query time. One factor is the retrieval of m MAP21-subtrees, the second factor is the merge operations of n page lists whose entry is the keyword and its corresponding MAP21-tree. The third factor is the time to read the m pages lists from disk. The pages in a page list whose entry is a primary-keyword is a subset of pages in the page list whose entry is the corresponding keyword of primary time period, so the length of a page list whose entry is a primary-keyword is reduced smaller than the above structure.

3.4. Expanded Inverted File

Inverted file is built by B+-tree, the leaf node of B+-tree points to the inverted list of the keyword. Because the index structure of most related work about temporal text indexing is usually based on inverted file index. The basic lookup operation in text-indexing is to retrieve the document identifiers of all document s that contains a particular word w. we consider designing the similar structure. The difference is the details of the page list pointed by the leaf node of B+-tree for inverted file. In the page list, there is not only corresponding set of URLs of Web pages but also the corresponding update time and the primary time period of one Web page. As shown in Fig.6.

The storage in disk includes the page lists and one B+-tree.

So, $Storage_4 = B^4_{Tree} + B^4_{List} = O(K) + O\left(\sum_{t=1}^{T} P_T(t)\right)$

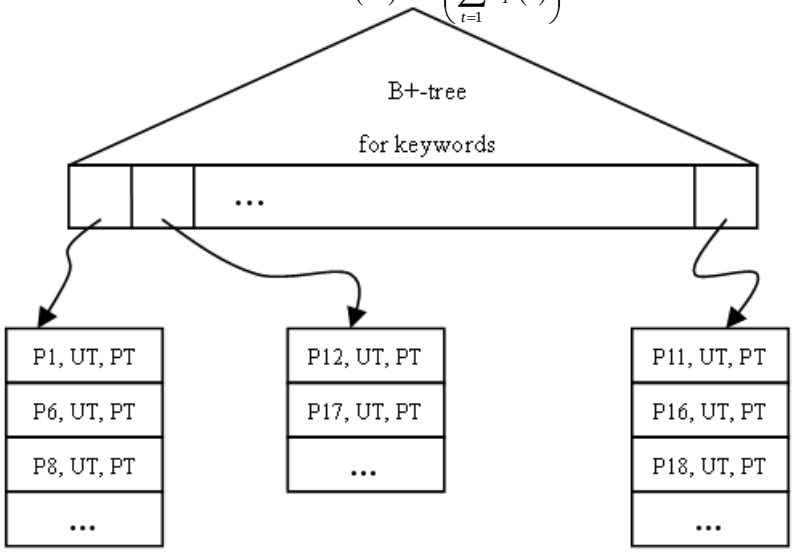


Fig. 6. Illustration of expanded inverted file structure(UT = Update Time, PT = Primary Time)

The main cost of storage in disk is caused by the total length of page lists whose entry is a keyword.

Assume there is the input of *m* keywords, one update time instant and a primary time period. The online computation includes: (1) the retrieval of the m keywords and the loading of corresponding page lists from disk; (2) the merge of the m page lists filtrated the time period not included in and error time instant in memory. So,

$$Time_4 = T_{Tree}^4 + T_{I/O}^4 + T_{mg}^4$$
$$= T_{Tree}(K) + \left(\sum_{i=1}^{m} T_{disk} \cdot O\left(P_T(t_i) / B_{section} \right) \right) + O\left(\sum_{i=1}^{m} P_T(t_i) \right)$$

There are also main factors for the query time. One factor is the merge of m page lists, at the same time, to filtrate the Web pages included wrong time information that users do not want. The other factor is the time to read the page lists from disk.

3.5. First MAP21-tree Then Inverted File

As shown in Fig.7, a MAP21-tree is built on all the primary time periods of Web pages. The leaf node of the tree points to a B+-tree containing the keywords that are included in the corresponding time periods. The leaf node of every B+-tree points to a page list. Each page list contains the update time of Web pages and the corresponding set of URLs.

The main storage in disk includes the page lists, B+-trees pointed by P primary times and one MAP21-tree. So,

$$Storage_5 = B_{Tree}^5 + B_{List}^5$$
$$= P \cdot O(K) + O(P) + O\left(\sum_{t=1}^{T} P_T(t) \right)$$

The main cost of storage in disk is caused by the total length of page lists whose entry is a putime-keyword.

If we input *m* keywords, one primary time period and a update time instant, the online computation has three parts: (a) searching the MAP21-tree and getting *n* primary time periods; (b) for each time period, retrieving the *m* keywords through the invert file, and the loading the corresponding page lists; (c) merging the *m* page lists. So,

$$Time_5 = T_{Tree}^5 + T_{I/O}^5 + T_{mg}^5$$
$$= m \cdot T_{Tree}\left(\overline{K} \right) + T_{Tree}(P) + \left(\sum_{i=1}^{m} T_{disk} \cdot O\left(P_T(t_i) / B_{section} \right) \right) + O\left(\sum_{i=1}^{m} P_T(t_i) \right)$$

In the above equation, $\overline{K}$ is the average number of leaf nodes in the B+-trees.

There are three factors that affect the query time. One factor is the retrieval of *m* B+-subtree, the second factor is the merge operations of page lists. The third factor is the time to read the *n* pages lists from disk.

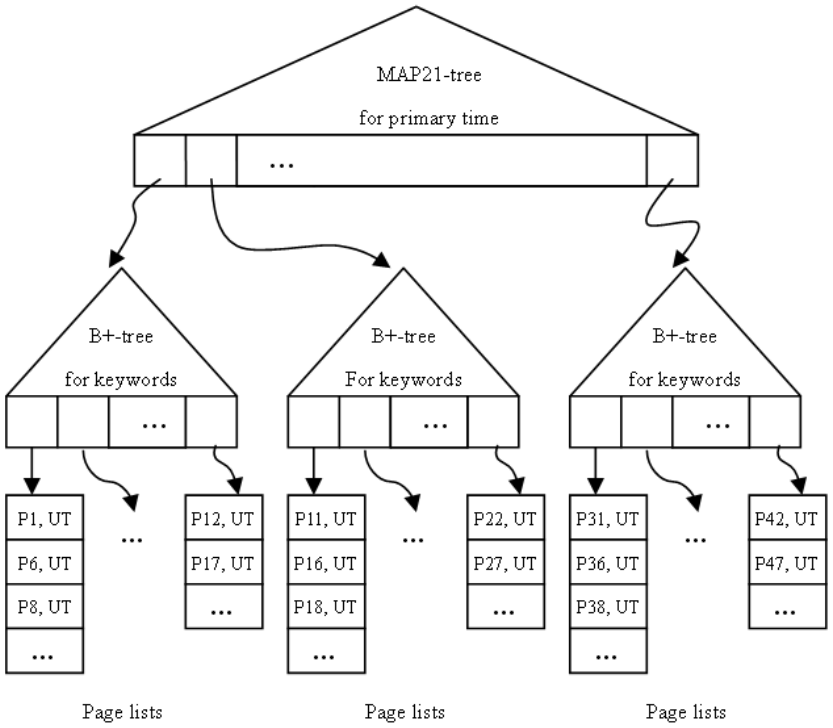


Fig. 7. Illustration of first MAP21-tree then inverted file (UT = Update Time)

There are some major differences between this structure and the index structure in Section 3.3, which refers to first inverted file then MAP21-tree. In the case that the index structure is first inverted file then MAP21-tree, if we input one keyword, one update time and one primary time period, we will first find the leaf node of keyword from one B+-tree, then retrieve one corresponding MAP21-tree to get right page lists from disk. If the index structure is first MAP21-tree then inverted file, the first step is finding *m* leaf nodes of primary time from one MAP21-tree according to the given primary time period, then retrieve each corresponding B+-tee to get right page lists from disk. Hence, the main difference between these two structures is the number of B+-subtree that is read from disk. The number of B+-subtrees will increase with the increasing of the range of primary time query.

4. Experiments

In this section, we will implemented the five hybrid index structures and make comparison experiment to show the different performance of those index structures under different workloads. For simplicity, we use the following notation to represent the five hybrid index structures discussed in the previous

section:

Structure 1: inverted file, B+-tree and MAP-21 triple index.
Structure 2: first inverted file then MAP21-tree and B+-tree.
Structure 3: first inverted file then MAP21-tree.
Structure 4: expanded inverted file.
Structure 5: first MAP21-tree then inverted file.

We mainly evaluate the query performance of the five index structures. In order to get a comprehensive result, we use five types of queries in the experiment. Those queries are:

TYPE 1: keyword query
TYPE 2: keywords + update time instant
TYPE 3: keywords + primary time instant
TYPE 4: keywords + update time instant + primary time instant
TYPE 5: keywords + update time period + primary time period

In our experiments, we focus on the index size, I/O number, and run time of each index structure under given workload and temporal-textual queries. Run time contains I/O number and memory time, and I/O time costs much longer than memory time. So we not only compare the run time but also compare the I/O number.

We used two types of dataset in our experiment: a simulated dataset and a real dataset. A simulated dataset is more regular than a real dataset, so the experiment on simulated dataset may achieve the desired result. A real dataset is more complex, so we experiment on it beside the simulated dataset. In the following part, we will describe the experimental results on simulation and real dataset respectively.

4.1. Simulation Experiment

4.1.1 Settings and Dataset

In our simulation experiment, we manually generate a simulated dataset. In this dataset, each Web page has one update time instant, one primary time period, five different keywords. All the time included in the dataset is limited in one year, i.e., the update time is set as a day in 2009, and the primary time is set as a time period in 2009. For simplicity, we use a number ranging from 1 to 100000 to represent a keyword. As a result, a generated record representing a Web page is as follows:

<URL, UT, PTs, PTe, key1, key2, key3, key4, key5>

Here, *UT* represents the update time, *PTs* and *PTe* represent the start and end date of primary time.

In our experiment, we generate three simulated datasets. They contain 1095 thousand records, 1825 thousand records, and 2555 thousand records, respectively. In the following text, we use T1095, T1825, and T2555 to indicate the three types of simulated datasets. We want to know whether the

result may change with the incremental datasets.

We run our experiment in a computer with an Intel Core 2.00 GHz CPU, 2 GB RAM, using Microsoft Window 7.

4.1.2 Comparison of Five Hybrid Index Structures

First, we compare the index size (Mbytes) of five hybrid index structures. We generate three trace, we show them respectively in Table 2, Table 3 and Table 4.

Table 2, Table 3 and Table 4 show that Structure 1 has the biggest total index size. The size of Structure 2 is bigger than that of Structure 4 but smaller than that of Structure 1, and Structure 3 and Structure 5 has smaller size than Structure 4. So the index size of Structure 3 and Structure 5 is the smallest in the five index structures.

Table 2. Index size for five hybrid index structures under the T1095 dataset (MBytes)

	Keyword index	Update time index	Primary time index	Total index size
Structure 1	684.62	136.86	136.86	958.34
Structure 2	0	136.86	700.96	837.82
Structure 3	0	0	759.78	759.78
Structure 4	802.25	0	0	802.25
Structure 5	759.81	0	0	759.81

Table 3. Index size for five hybrid index structures under the T1825 dataset (MBytes)

	Keyword index	Update time index	Primary time index	Total index size
Structure 1	1164.86	228.13	228.13	1612.12
Structure 2	0	228.13	1191.67	1419.80
Structure 3	0	0	1289.69	1289.69
Structure 4	1360.90	0	0	1360.90
Structure 5	1289.74	0	0	1289.74

Table 4. Index size for five hybrid index structures under the T2555 dataset (MBytes)

	Keyword index	Update time index	Primary time index	Total index size
Structure 1	1621.13	319.38	319.37	2259.88
Structure 2	0	319.37	1657.89	1977.26
Structure 3	0	0	1795.14	1795.14
Structure 4	1895.58	0	0	1895.58
Structure 5	1795.21	0	0	1795.21

Second, we compare page I/O numbers of the five index structures on the basis of the five types of queries mentioned before. The page size is set as 4 Kbytes. The total I/O number of every query includes the retrieval of tree nodes and page lists from disk. We generate 300 queries randomly, each of which contains three keywords and one update time and one primary time, and calculate the average I/O number of the 300 queries. The results are shown in Table 5, Table 6 and Table 7. Additionally, the reason why we choose three words is that a users' query usually contains one to three keywords in the Web search environment.

Table 5. Page I/O# for five hybrid index structures under the T1095 dataset

	Type 1	Type 2	Type 3	Type 4	Type 5
Structure 1	1687	1779	1719	1811	2052
Structure 2	1690	1782	40	131	366
Structure 3	1695	1695	40	40	95
Structure 4	1701	1701	1701	1701	1701
Structure 5	8252	8252	37	37	107

Table 6. Page I/O# for five hybrid index structures under the T1825 dataset

	Type 1	Type 2	Type 3	Type 4	Type 5
Structure 1	2550	2707	2604	2762	3185
Structure 2	2554	2712	59	218	629
Structure 3	2561	2561	59	59	159
Structure 4	2571	2571	2571	2571	2571
Structure 5	9105	9105	56	56	171

Table 7. Page I/O# for five hybrid index structures under the T2555 dataset

	Type 1	Type 2	Type 3	Type 4	Type 5
Structure 1	3467	3687	3542	3762	4351
Structure 2	3473	3694	79	300	873
Structure 3	3482	3482	79	79	217
Structure 4	3495	3495	3495	3495	3495
Structure 5	10126	10126	74	74	227

In Table 5, Table 6 and Table 7, we can see the number of blocks read from disk based on Structure 1, 2, 3 and 4 is similar for Type 1. The number of blocks based on Structure 3 and Structure 4 is smaller than the other three structures for Type 2, and the two structures have almost no difference or little difference. For Type 3, Structure 2, Structure 3 and Structure 5 have obvious advantages over the other two structures. Structure 3 and Structure 5 have good performance for Type 4 when reading blocks from disk. Structure 3 and

Structure 5 are better than the other structures for Type 5. In addition, Structure 3 is little better than Structure 5 as the increased number of Web pages.

Table 5 to Table 7 show that Structure 3 has the smallest I/O cost for five types of queries, Structure 5 is better for three types of queries except Type 1 and Type 2. Structure 3 and Structure 5 both have some subtrees. The advantage of subtree is that it filters some unmatched Web pages according to the given keywords or primary time.

Third, we compare the run time of five structures for five types of queries. Run time is calculated from the input of query to the output of the right URL set. The queries are the same above. The results are shown in Table 8, Table 9 and Table 10. The unit of time is second.

Table 8. Run time for five hybrid index structures under the T1095 dataset (seconds)

	Type 1	Type 2	Type 3	Type 4	Type 5
Structure 1	2.15	2.35	2.34	2.37	2.36
Structure 2	2.21	2.48	0.06	0.17	0.45
Structure 3	2.33	1.29	0.07	0.07	0.16
Structure 4	2.24	1.26	1.38	1.37	1.66
Structure 5	9.31	8.28	0.06	0.06	0.18

Table 9. Run time for five hybrid index structures under the T1825 dataset (seconds)

	Type 1	Type 2	Type 3	Type 4	Type 5
Structure 1	3.10	3.18	3.17	3.21	3.26
Structure 2	3.15	3.27	0.08	0.25	0.71
Structure 3	3.33	2.05	0.10	0.09	0.25
Structure 4	3.16	1.95	2.08	2.09	2.17
Structure 5	10.30	9.19	0.09	0.09	0.24

Table 10. Run time for five hybrid index structures under the T2555 dataset (seconds)

	Type 1	Type 2	Type 3	Type 4	Type 5
Structure 1	4.05	4.23	4.21	4.22	4.30
Structure 2	4.48	4.62	0.13	0.44	1.30
Structure 3	4.22	3.17	0.14	0.14	0.39
Structure 4	3.52	2.63	2.80	2.80	2.93
Structure 5	11.21	9.90	0.11	0.11	0.34

Run time includes three main parts. They are the time for retrieving corresponding trees, the time for reading page lists from disk and the time for merging page lists. The run time in the first and the second part is mainly determined by the page I/O numbers listed in Table 5 to Table 7. The time for

merging page lists is much smaller than the time for reading page lists from disk. So we can get the similar comparison result as shown in Table 5 to Table 7. We can see the run time based on Structure 1, 2, 3 and 4 is similar for Type 1. Structure 3 and Structure 4 is faster than the other three structures for Type 2. In addition, these two structures have almost no difference or little difference in run time for Type 2. Structure 2, Structure 3 and Structure 5 have obvious advantages over the other two structures for Type 3. Structure 3 and Structure 5 have good performance for Type 4. Structure 3 and Structure 5 are better than the other structures for Type 5. Additionally, Structure 3 is little better than Structure 5 as the increased number of Web pages. Generally, Structure 3 wins the best in the measurement of run time.

4.1.3 Rebuilt Time of Five Hybrid Index Structures

We consider the update cost now. When Web pages update are small, update pages can be inserted into the specified location directly. Once the Web pages update are large, rebuild index is a good choice. We compare rebuild cost of five hybrid index structures here. The result of three different simulated datasets is shown in Table 11.

Table 11. Rebuild time for five hybrid index structures under the T1095, T1825 and T2555 dataset (seconds)

	T1095	T1825	T2555
Structure 1	1995.18	7249.17	11465.50
Structure 2	4230.64	8682.05	12781.80
Structure 3	4877.58	8641.93	14312.10
Structure 4	3891.86	8180.55	12537.10
Structure 5	4097.65	7713.05	10610.20

Table 11 shows that the rebuild time of Structure 1 costs least under the three simulated datasets. Structure 4 and Structure 5 cost longer than Structure 1, and the rebuild time of Structure 4 and Structure 5 has little difference. Structure 2 and Structure 3 cost longest. Although the rebuild time of Structure 3 cost longest, its query performance is very dominant, so we can ignore the rebuild cost.

4.2. Experiment on real dataset

4.2.1. Settings and Dataset

We choose the real dataset from the corpus of SouGou lab (http://www.sogou.com/labs/) which records games, sports, IT, domestic and

international news in May 2008 from some news sites.

The experiment on real dataset is more complicated than the simulation experiment. In this experiment, we simply describe how to exact the update time and the primary time in one real Web page. We use the real dataset from news Web pages, and the news pages have their own characteristics. In the news, publish time is usually as the update time and primary time is often appears in the first paragraph. The exaction of primary time is more complicated than update time. If there is one time instant in the first paragraph, we consider it as the primary time; If there is two or more time instants in the paragraph, we consider choosing the nearest instant to update time instant as primary time; If there are time instants and time periods, we first choose the time period as the primary time of the Web page.

Keywords of every Web page are exacted by a tool called ICTCLAS (http://ictclas.org/), which is the most efficient tool for the Chinese words segmentation. Each word is mapped a value in memory by ELFHASH function.

We use 250 thousand news Web pages as our real dataset and extract approximately 210 thousand different keywords. The experimental computer environment is the same as that in the simulation experiment.

4.2.2. Comparison of Five Hybrid Index Structures

We still use the five query types to measure the index size. Page I/O#, and run time of each hybrid index structure.

First, we compare the index size (MBytes) of five hybrid index structures. The results are shown in Table 12.

Table 12. Index size for five hybrid index structures under the real dataset (MBytes)

	Keyword index	Update time index	Primary time index	Total index size
Structure 1	1307.69	31.25	31.25	1370.19
Structure 2	0	31.25	1315.04	1346.29
Structure 3	0	0	1425.14	1425.14
Structure 4	1528.07	0	0	1528.07
Structure 5	1425.22	0	0	1425.22

Table 12 shows that Structure 1 has the largest size, and Structure 2 is worse than Structure 4 but better than Structure 1. Besides, the index size of Structure 3 and Structure 5 is the smallest in the five index structures. Therefore, we have the same conclusion as what we got in the simulation experiment.

Second, we compare page I/O# of five index structures for five query types. The page size is set as 4 KBytes. We choose 300 queries of three keywords from the search log, one update time and one primary time randomly and calculate the average of the 300 queries. We show them in Table 13.

Table 13. Page I/O# for five hybrid index structures under the real dataset

	Type 1	Type 2	Type 3	Type 4	Type 5
Structure 1	481	745	738	1001	2950
Structure 2	485	749	27	291	1780
Structure 3	486	486	27	27	67
Structure 4	485	485	485	485	485
Structure 5	3788	3788	26	26	79

Table 13 shows that the number of blocks read from disk based on Structure 1, 2, 3 and 4 is similar for Type 1. The number of blocks based on Structure 3 and Structure 4 is smaller than those of the other three structures for Type 2 and the two structures have almost no difference or little difference. For Type 3, Structure 2, Structure 3 and Structure 5 have obvious advantages over the other two structures. Structure 3 and Structure 5 have good performance for Type 4 when reading blocks from disk. Structure 3 and Structure 5 are better than the other structures for Type 5, in addition, Structure 3 is little better than Structure 5 as the increased number of Web pages.

As a result, Structure 3 is the best index structure for five types of query, Structure 5 is better for three types of queries except Type 1 and Type 2. This is the same as what we get in the simulation experiment.

Third, we compare the run time of five structures for five types of search. Run time is the process which is from the input of query user need to the output of the right URL set. The queries are the same above. We show them in Table 14. The unit of time is second.

Table 14. Run time for five hybrid index structures under the real dataset (seconds)

	Type 1	Type 2	Type 3	Type 4	Type 5
Structure 1	0.96	1.30	1.31	1.62	3.85
Structure 2	1.03	1.34	0.08	0.41	2.14
Structure 3	1.12	0.53	0.08	0.08	0.14
Structure 4	1.11	0.54	0.60	0.58	0.59
Structure 5	6.00	5.30	0.08	0.08	0.15

4.2.3. Rebuilt Cost of Five Hybrid Index Structures

We consider the update cost now. When Web pages update are small, update pages can be inserted into the specified location directly. Once the Web pages update are large, rebuild index is a good choice. We compare rebuild cost of five hybrid index structures under the real dataset. The result is shown in Table 15.

Table 15. Rebuild time for five hybrid index structures under the real dataset (seconds)

	Structure 1	Structure 2	Structure 3	Structure 4	Structure 5
Rebuilt time	9146.88	29235.10	31310.14	14170.90	3829.13

Table 15 shows that the rebuild time of Structure 5 costs least, and Structure 3 has the longest rebuild time. Although the rebuild cost of Structure 3 is larger, compare with the query performance, the rebuild cost is affordable.

5. Related Work

5.1. Temporal Information Extraction and Retrieval

Traditional commercial search engines, such as Google, Bing, and Baidu, have noticed the value of temporal information in Web search. They all provide some ways for users to perform a Web search based on time. E.g. Google uses the daterange option to express a temporal predicate. However, those commercial search engines only support the crawled dates of Web pages, i.e., users can only query Web pages towards their creation dates in database. There are also some other temporal information retrieval systems which use similar methods as Google to process temporal information of Web pages, such as Goo [4], Infoseek [5], Namazo [6], Chronica [7], and so on. Generally, there is a gap between the crawled time and content time of a Web page. For instance, if a news page reports that in Aug 8th, 2008 the Olympic Games will be held in Beijing, China, but it is posted and crawled in Jun. 21st, 2006. In such case, “Jun. 21st, 2006” will be regarded as the temporal information of this news page, but unfortunately it does not report the right temporal information of the page. To our knowledge, there are few search systems considering the temporal information embedded in Web pages [1, 7]. The system presented in [8] aims at extracting and indexing the content time of Web pages, but it only considered the business hour extraction, and can not deal with implicit time such as “today”, “Tuesday”, “Last Christmas”, and so on. Other work in Natural Language Processing (NLP) focused on temporal information extraction and annotation from text. There are a lot of tools which can extract temporal elements from text. Many of them are towards English text, such as Lingua::EN::Tagger [9] and TempEX [10]. In recent years, some temporal extraction tools for non-English languages were also presented. For example, KTX was a temporal information extracting tool for Korean text [11], and CTEMP was for Chinese text [12]. Most of temporal information extraction tool in NLP are based on the temporal information annotation standard TIMEX2 [13] or TimeML [14]. Though temporal

information extraction in NLP is relative mature, little effort has been done to combine them into Web search engines.

5.2. Temporal-Textual Indexes

The current temporal text indexing is mainly to the versioned document collections such as Web archives [15, 16]. There have been some indexing approaches on directly addressing the issue of temporal-textual indexing. Anick and Flynn [17] have pioneered this research to support versioning in a full-text index on bitmaps for terms in current versions, and delta change records to track incremental changes to the index backward over time. The disadvantage is the costly recreation of previous states. Recent work in [19, 23-26] and their earlier proposals concentrate on the problem of supporting text-containment queries and neglect the relevance scoring of results. Stack [20] reports practical experiences made when adapting the open source search engine Nutch to search Web archives. Weikum et al. address the temporal dimensions completely by extending the inverted files index to make it ready for temporal search and implement the time-travel text search in the FluxCapacitor prototype [27, 28]. In contrast, research in temporal databases has produced several index structures tailored for time-evolving databases. A comprehensive overview of the state-of-art is available in [29]. Unlike the inverted file index, their applicability to text search is not well understood.

Temporal indexes have been deeply studied in temporal database area. In temporal database, two dimensions of time, which are valid time and transaction time, may be considered in the index [21]. Therefore, R-tree [22] and its variation as the access structures for spatial data may also be used as a temporal index. Among all the temporal indexes proposed before, the MAP21-tree [3], which utilizes standard B+-trees, provides efficient indexing of valid time period, and has good performance in time instant query and range query. The idea of MAP21-tree is to map a two-dimensional time period to a one-dimensional number and then to use a B+-tree to build the index structure. So in this paper we choose the MAP21-tree as the basic temporal index structure.

The index structure of most related work about temporal-textual indexing is usually based on inverted file index [18]. However, the main difference between our work and previous researches is that we consider to index both update time and content time for Web pages, while previous temporal-textual indexes usually focused on indexing update time, because they are designed for Web archive system or document versioning.

In addition, in this paper we introduce the concept of primary time. Hence, the temporal-textual index structures studied in this paper involve keywords, update time, and content time. Traditional models in information retrieval have been widely studied since 1970s, among which the Vector Space Model (VSM) [15, 16] and the Probabilistic Relevance model (PPR) [16, 17, 20] are two representatives. In VSM, all the keywords are represented into high dimension feature vectors rather than representing the keywords with binary

value. The problem of VSM is it considers little about the relationship between keywords. The BM25 model, as a popular Probabilistic Relevance model [17], ranks Web pages based on their probability of relevance with the query. This model needs to know the information about which Web pages are relevant with the query, which is very difficult to realize in a large dataset.

Pagerank ranking model [21] is an offline ranking algorithm which is based on the number of Web pages that are linked by other pages in the whole Web, and the quality of the sources of the links. The problem of Pagerank is it only considers links but ignores the similarity between the query and Web pages.

Most of time-related Web search now concentrates on Web archive system [22-24]. A Web archive system is used to store and manage historical Web pages and then provides evolutional information of the Web. The history of a Web page is typically captured by the versioning technique, i.e., the new version of a Web page is stored with an explicit update timestamp. However, Web archive systems only consider the update timestamps of Web pages. They do not take into account the content time of Web pages, which is much different from the research scope of this paper.

In recent years, several researchers have studied ways to find fresh Web pages. The TimedPageRank algorithm [25] was proposed in a Web-based literature searching prototype. It uses the posted time of paper to perform the ranking process. If we map it into a general Web search engine, the posted time of paper can be regarded as the publication time or update time of Web page. It can not support queries focusing on the content time. In [26], a temporal search system for business hours was studied, which tried to answer such questions 'Which shops are open and in which time are they open'. In this system, the time granularity was restricted in hour, e.g., '9:00 AM'. Besides, it does not support implicit time, such as Christmas, the National Day. So it is not suitable for general Web search engines.

The language modeling approach for information retrieval was first in 1998 [5]. Its basic idea is to estimates the probability of the query given the language model of a Web page, and ranks Web pages according to those probabilities. There are also some variants of this approach [27, 28, 29]. Previous studies have shown that the language model has a better performance than traditional models such as VSM and BM25, and the experimental results in this paper also proved this truth.

There are also some recent works focusing on temporal language models [18, 19, 30], which integrate temporal information into the framework of language models. In the literature [19], a time-based language model was proposed, which emphasized that recent documents could be better to satisfy users' needs. However, this model only concentrates on the publication time of Web pages, and therefore is useful for querying recent events but is not a general framework. A similar work could be found in [30], which also focused on the publication time. The recent work in [18] integrated the content time of Web pages into the language model. In this approach, the researchers proposed the assumptions of temporal relevance, which first had a filter process to filter all the Web pages containing no temporal references in their contents, and then used the triangle distribution model to simulate the

probabilities of the querying time appearing in each Web page. However, this model does not consider the relationship between the keywords and the temporal references in the page content.

6. Conclusions

In this work we have designed and implemented five hybrid index structures for temporal-textual Web search and studied the performance of these index structures. We conduct a comprehensive experiment on both the simulated and real datasets, and use five temporal-textual query types to evaluate the index size, page I/O#, and run time of each hybrid index structure. Both the simulation and real experimental results show that the index structure "first inverted file then MAP21-tree" has the best performance among the five index structures. Therefore, it should be an acceptable choice for indexing temporal and text information in a temporal-textual Web search engine.

In the future research, we will focus on the update performance of the index structure, and integrate the hash policy to improve the update and search performance. Another work will be the compression of the index, since the index size of the hybrid index structure is still too big for Web search.

Acknowledgements. This work is supported by the National Science Foundation of China (no. 70803001), the Open Projects Program of National Laboratory of Pattern Recognition (20090029), the Key Laboratory of Advanced Information Science and Network Technology of Beijing (xdxx1005), and the USTC Youth Innovation Foundation.

References

1. Alonso, O., Gertz, M., Yates, R., On the value of temporal information in information retrieval, In Proc. Of SIGIR'07, 35-41. (2007)
2. Jensen, C., Temporal Database Management, PhD Thesis, http://www.cs.auc.dk/~csj/Thesis/. (2000)
3. Nascimento, M., Dunham, M., Indexing Valid Time Databases via B+-Trees. IEEE Transactions on Knowledge and Engineering. Vol.11(6), 929-947. (1999)
4. Goo, http://www.goo.ne.jp/
5. Infoseek, http://www.infoseek.co.jp/
6. Namazu, http://www.namazu.org/
7. Deniz, E., Chris, F., Terence, J., Chronica: a temporal Web search engine. In Proc. Of ICWE'06, 119-120. (2006)
8. Taro, T., Katsumi, T., Temporal and Spatial Attribute Extraction from Web Documents and Time-Specific Regional Web Search System, in Proc. Of W2GIS'04, 14-25. (2004)
9. Lingua::EN::Tagger, http://search.cpan.org/~acoburn/ Lingua-EN-Tagger
10. Mani, I., Wilson, G., Robust temporal processing of news. In Proc. of ACL'00, 69-76. (2000)

11. Jang, S., Baldwin, J., Mani, I., Automatic TIMEX2 tagging of Korean news, ACM Trans. Asian Lang. Inf. Process, Vol.3(1), 51-65. (2004)
12. Wu, M., Li, W., Lu, Q., Li, B., CTEMP: A Chinese Temporal Parser for Extracting and Normalizing Temporal Information. In Proc. Of IJCNLP'05, 694-706. (2005)
13. TIMEX2. In http://timex2.mitre.org/
14. TimeML. In http://www.timeml.org/site/index.html
15. Hersovici, M., Lempel, R., Yogev, S., Efficient Indexing of Versioned Document Sequences. In Proc. of ECIR. (2007)
16. Grandi, F., Introducing an Annotated Bibliography on Temporal and Evolution Aspects in the World Wide Web. SIGMOD Record 33(2), 84-86. (2004)
17. Anick, P., Flynn, R., Versioning a Full-Text Information Retrieval System. In Proc. Of SIGIR. (1992)
18. Zobel, J., Moffat, A., Inverted Files for Text Search Engines. ACM Comput. Surv., Vol.38(2), 6. (2006)
19. Nørvag, K., Nybø, A., DyST: Dynamic and Scalable Temporal Text Indexing. In Proc. Of TIME. (2006)
20. Stack, M., Full Text Search of Web Archive Collections. In Proc. Of IWAW. (2006)
21. Kumar, V., Tsotras, J., Faloutsos, C., Designing Access Methods for Bitemporal Databases. IEEE Trans. Knowl. Data Eng, Vol.10(1), 1-20 (1998)
22. Beckmann, N., Kriegel, H., Schneider, R., Seeger, B., The R-tree: An efficient and robust access method for points and rectangles. In Proc. Of SIGMOD, 322-331. (1990)
23. Nørvag, K., Space-Efficient Support for Temporal Text Indexing in a Document Archive Context. In Proc. Of ECDL, 511-522. (2003)
24. Nørvag, K., Supporting temporal text-containment queries in temporal document databases. Data Knowl. Eng., Vol.49(1), 105-125. (2004)
25. Nørvag, K., Nybø, A., Improving Space-Efficiency in Temporal Text-Indexing. In Proc. Of DASFAA, 791-802. (2005)
26. Nørvag, K., Nybø, A., DyST: Dynamic and Scalable Temporal Text Indexing. In Proc. Of TIME, 204-211. (2006)
27. Berberich, K., Bedathur, S., Neumann, T., Weikum, G., FluxCapacitor: Efficient Time-Travel Text Search. In Proc. Of VLDB, 1414-1417. (2007)
28. Berberich, K., Bedathur, S., Neumann, T., Weikum, G., A Time Machine for Text Search. In Proc. Of SIGIR, 519-526. (2007)
29. Salzberg, B., Tsotras, V., Comparison of Access Methods for Time-Evolving Data. ACM Comput. Surv., Vol.31(2), 158–221. (1999)
30. Li, W., Wong, K., Yuan, C., A Model for Processing Temporal References in Chinese, In Proc. of Workshop on Temporal and Spatial Information Processing at ACL'01, 33-40 (2001)
31. Alonso, O., Gertz, M., Yates, R., Clustering and exploring search results using timeline constructions. In Proc. Of CIKM, 97-106. (2009)
32. Jin, P., Chen, H., Zhao, X., Yue, L., Hybrid Index Structures for Temporal-textual Web Search, In Proc. of APWeb, LNCS 6612, 271-277, (2011)

Peiquan Jin is an Associate Professor at the School of Computer Science and Technology, University of Science and Technology of China (USTC). He received his Ph.D. degree in computer science from University of Science and Technology of China (USTC) in 2003. After that he spent two years for his post-doc research in the Department of Electronic Engineering & Information Science, USTC. He was a Visiting Scientist at the University of Kaiserslautern, Germany, in 2009. His research interests include Web search, knowledge management, and databases. Dr. Jin is a member of ACM, and is an editor of International Journal of Information Processing and Management and International Journal of Knowledge Society Research. He serves as a PC member of many international conferences, including DEXA'09-11, WAIM'11, NCM'08-11, and NDBC'09-11.

Hong Chen is currently a third-year master student at the School of Computer Science and Technology, University of Science and Technology of China. Her main research interests include Web information extraction and retrieval.

Xujian Zhao is currently a PhD candidate at the School of Computer Science and Technology, University of Science and Technology of China. His research interests are focused on temporal information retrieval in the Web, especially on Web-based news search. He spent six months in Alibaba in 2010 for e-business data mining work, and was a visiting scientist in University of Saarland, Germany, in 2011. He has published about ten papers in peer-reviewed conferences and journals.

Xiaowen Li is currently a third-year master student at the School of Computer Science and Technology, University of Science and Technology of China. Her main research interests include temporal information extraction and time-aware ranking for Web pages.

Lihua Yue is a full professor at the School of Computer Science and Technology, University of Science and Technology of China. She received his master degree in computer science in 1994 from University of Science and Technology of China. Her research interests are mainly concentrated on database and information systems. She is a member of ACM and a senior member of China Computer Federation (CCF). She has published more than 50 papers in peer-reviewed conferences and journals.

Received: April 07, 2010; Accepted: March 14, 2011.

DOI: 10.2298/CSIS100907026S

Parallel Processing on Block-based Gauss-Jordan Algorithm for Desktop Grid

Yizi SHANG[1], Guiming LU[2], Ling SHANG[2,3], and Guangqian WANG[1]

[1] State Key Laboratory of Hydro-science and Engineering, Tsinghua University, Beijing, China
syz05@mails.tsinghua.edu.cn
[2] North China University of Water Conservancy and Electric Power, School of Information Engineering, Zhengzhou, China
njshangl@gmail.com
[3] Lifl, University of Science and Technology of Lille, Lille, France
ling.shang@lifl.fr

Abstract. Two kinds of parallel possibilities exist in the block-based Gauss-Jordan (BbGJ) algorithm, which are intra-step and inter-steps based parallelism. But the existing parallel paradigm of BbGJ algorithm just aiming at the intra-step based parallelism, can't meet the requirement of dispatching simultaneously as many tasks as possible to computing nodes of desktop grid platform exploiting thousands of volunteer computing resources. To overcome the problem described above, this paper presents a hybrid parallel paradigm for desktop grid platform, exploiting all the possible parallelizable parts of the BbGJ algorithm. As well known to us all, volatility is the key issue of desktop grid platform and faults are unavoidable during the process of program execution. So the adapted version of block BbGJ algorithm for desktop grid platform should take the volatility into consideration. To solve the problem presented above, the paper adopts multi-copy distribution strategy and multi-queue based task preemption method to ensure the key tasks can be executed on time, thus ensure the whole tasks can be finished in shorter period of time.

Keywords: Gauss-Jordan algorithm, desktop grid, data dependence, parallelism, hybrid parallel paradigm.

1. Introduction

A good parallel programming paradigm should help to maximize parallel execution of the algorithm, thus achieving better performance. And the choice of paradigm is determined by the available parallel computing resources and by the type of parallelism inherent in the problem [14].

Most of the scientific communities have the desire to minimize economic risk and rely on consumer based off-the-shelf technology. Exploiting the significant computational capability available in the internet-based desktop

Grid environment has gained an enthusiastic acceptance within the high performance computing community [10,18] and Desktop Grid computing has been recognized as the wave of the future to solve large scientific problems, especially in the days of multi-core architectures [12] are widely used in personal computer. Paper [19] evaluates the potential capability of desktop grid systems. With more and more personal computers with multi-core architectures join the desktop grid platform, the platform can provide huge process However, realizing better performance of the parallel algorithm in desktop grid platform requires exploiting as more thread-level parallelism as possible and avoiding the influence from computing nodes joining in (leaving from) the platform without any advance notice. It is important to find a solution to get maximal parallelism available to an algorithm and find appropriate schedule mechanism for the paltform, thus achieving its better performance in the desktop Grid platform.

BbGJ algorithm [1][2][4], as a classical method of large scale matrix inversion, can be used in hydro-science, weather prediction, aircraft design, graphic transformation and so on. Though there are many approaches on large scale matrix inversion such as SCALAPACK library, GOTO library. But these kinds of methods aren't available to desktop grid paradigm. The reason is that it is impossible to install a special library on each computing nodes. It is meaningful to make further research on BbGJ algorithm. Serge shows its parallel version adapting to MIMD [1]. Melab et al not only give us its parallel version tailoring to MARS but also analyze all the possible parallelism in the algorithm [2][4]. Aouad et al present its adapted version for grid platform [3, 7]. But these programming paradigm given by Melab and Aouad doesn't take inter-steps parallelism into consideration. More important, they can't be adapted to desktop grid environment for the special characters of desktop grid systems.

To improve the efficiency of the algorithm, it is important and necessary to find an approach which can exploit all the inter-steps and intra-step based parallelism in the algorithm. Thus more tasks can be generated and executed simultaneously on different computing nodes. Some characters of sequential BbGJ can be summarized as follows:

1) The number of iterative step of sequential BbGJ algorithm is equal to that of square root of matrix blocks partitioned.

2) In the BbGJ algorithm, the parts of available parallel granularity are based on blocks (which are sub-matrices partitioned). data dependence between blocks plays an important role in deciding which parts of program can be executed concurrently.

3) All the data input and output are based on blocks and the number of data input is same in each iterative step of BbGJ algorithm.

4) The object of data input on blocks is regular. Operation '1','3','5' (marked in sequential BbGJ algorithm in section 2) work on the matrix A while Operation '4','6','7' work on the matrix B which is the inversion of matrix A (see Fig.6).

From above, we know data dependence between different blocks is the key factor to solve the problem presented above. To the convenient of making

research on data dependence between blocks, we regard the two matrix (original matrix and its inverted matrix) as an augmented matrix. Through the analysis on executive process of BbGJ algorithm, we find that a set of three-tuples (k,i,j) can be used to control all the data dependence in the process of algorithm execution. In which, k represents the number of iterative step of BbGJ algorithm, i represents the row of augmented matrix while j is the column of the matrix. So those three-tuples are used as condition to control when and which parts can be executed. At the same time, we can find and mark the key path of depended tasks and the key path can be described using a series of three tuples. Based on that, this paper presents a hybrid programming paradigm, considering all the possible parallelism in the algorithm and appropriate allocation strategy.

The remainder of the paper is organized as follows: The data dependence in BbGJ will be described in the section two. Section three will show us the formal description of BbGJ The fourth section will present us a hybrid parallel programming paradigm for desktop grid system and key path of BbGJ using a series of three-tuples. The evaluation on hybrid paradigm will be made in section five. Section six is the conclusion and future works.

2. Data Dependence in the Algorithm

Let *A* and *B* be two dense matrices of dimension *N*, and let *B* be the inverted matrix of *A*, i.*e. AB=BA=I.* Let *A* and *B* be partitioned into a matrix of *q×q* blocks of dimension n which *n=N/q.* The sequential block-based Gauss-Jordan algorithm is the following:

Algorithm 1

Input: $A,B \leftarrow I\ (n,n),\ q$

Output: $B=A^{-1}$

For k=1 to q do // we call this as “iterative”

$A^{k}_{k,k} \leftarrow (A^{k-1}_{k,k})^{-1}$ (1)

$B^{k}_{k,k} \leftarrow A^{k}_{k,k}$ (2)

For j=k+1 to q do // we call this as “loop”

$A^{k}_{k,j} \leftarrow A^{k}_{k,k} A^{k-1}_{k,j}$ (3)

End For

For j=1 to k-1 do // we call this as “loop”

$B^{k}_{k,j} \leftarrow A^{k}_{k,k} B^{k-1}_{k,j}$ (4)

End For

For j=k+1 to q do // we call this as “loop”

For i=1 to q and $i \neq k$ do

$A^{k}_{i,j} \leftarrow A^{k-1}_{i,j} - A^{k-1}_{i,k} A^{k}_{k,j}$ (5)

End For

End For

For j=1 to k-1 do // we call this as “loop”
For i=1 to q and i≠k do
$B^{k}_{i,j} \leftarrow B^{k-1}_{i,j} - A^{k-1}_{i,k} B^{k}_{k,j}$ (6)
End For
End For
For i=1 to q and i≠k do // we call this as “loop”
$B^{k}_{i,k} \leftarrow - A^{k-1}_{i,k} A^{k}_{k,k}$ (7)
End For
End For

2.1. Parallelism in the Algorithm

From algorithm 1, we can find that each step of algorithm is made up of five loops (the third, fourth, fifth, sixth and seventh operation in algorithm 1) and two other operations (the first and second). At each step k (k = 1,..., q), the first operation is used to get the inverted matrices of sub-block matrices; the second operation executes the operation of assignment from the sub-block of matrix *A* to corresponding that of matrix *B*; the third operation computes the blocks belonging to the row of the pivot with index *j* is larger than *k* and the fourth operation computes the corresponding parts of matrix *B*; the fifth operation calculates the blocks of all columns of the matrix *A* with index i above and below that of the pivot row and the forth operation calculates the corresponding parts of matrix *B*; At last the seventh operation is used to compute the blocks of the column number *k* of matrix *B* except $B_{k,k}$.

The parallelization of the sequential BbGJ algorithm consists of exploiting two kinds of parallelism: the inter-steps parallelism and the intra-step parallelism. The intra-step parallelism consists mainly of exploiting the parallelism involved in each of the five loops (operation ‘3’ to operation ‘7’ in algorithm 1). It falls into two categories: the inter-loops parallelism and the intra-loop parallelism. Inter-steps parallelism is more complex because almost all the operations in algorithm 1 are restricted by the inter-steps data dependence. Then we will analyze those data dependence one by one.

2.2. Intra-step based data dependence in the Algorithm

We can describe the data dependence in the Algorithm1 using Fig.1 and Fig.2. In those figures the blue font represents the read operation. For example, the blue font ‘5’ in the first row, second column of matrix *A* in Fig.1 represents that: to this block, the first operation is ’5’ (see the number marked in algorithm 1) which is a read operation, i.e. operation ‘5’ will read this block as its input. The same way we can know that the black font ‘1’ in the second row, second column of matrix *A* in Fig 1 represents that: the operation ‘1’ (see the number marked in algorithm 1) will assign the value obtained by operation ’1’ to this block. Next we will take the block in the second row, first

column of matrix *B* in Fig.1 as example to show all the related operations executed in this block.

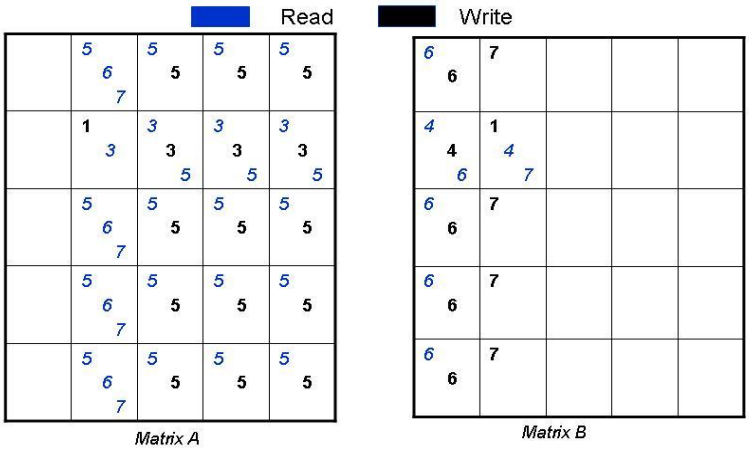


Fig.1. Blocks Operated at the Step 2 with q is 5 in Algorithm 1

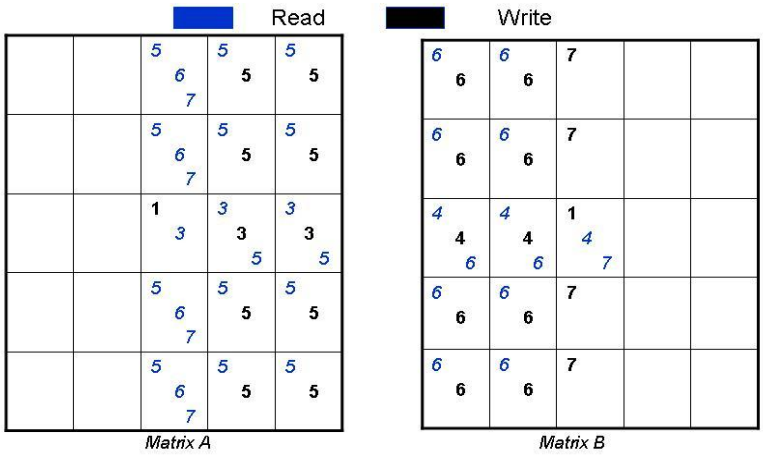


Fig.2. Blocks Operated at the Step 3 with q is 5 in Algorithm 1

From the Fig.1, we can know three operations executed in this block, of which blue fonts '4' and '6' are read operation and black font '4' is write operation. How those operations are executed and what's the sequence of those operations will be explained. To that block said above, operation will be executed from up to down, i.e. the read operation '4' will be executed first, after that write operation '4' begin to execute and the last operation is read operation '6'. After the finish of the three operations said above, no more operation on this block will be executed in this step. Fig.1 shows us all the

operations happened at the step 2 with *q* is 5 in the intra-step BbGJ algorithm. We can also get all the operations information at the step 3 with *q* is 5 from the Fig.2.

From Fig.1 and Fig.2 we can know that each of the q steps of the algorithm has the same number of write operation and the number is q^2. To the matrix *A* ,the write operation on it are operations '1', '3' and '5' while only operations '1', '6' and '7' act on matrix *B*. Through the analysis of Fig.1 and Fig.2, we also can get the data dependence in an iterative step of program execution. See Fig.3.

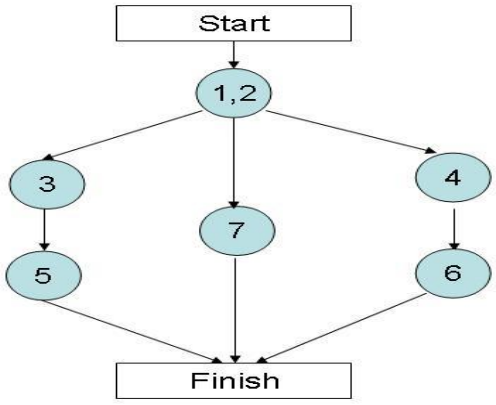


Fig. 3. The Data Dependence in a fixed iterative step

2.3. Inter-steps based data dependence in the Algorithm

We will take the data-dependence between step *k* and step *k+1* as example to show the inter-steps parallelism. Getting the value $A_{k,j}$ ($k=1…q$; $j=k+1…q$) (respectively $B_{k,j}$ in which $k=1…q$; $j=1…k-1$) of *k+1* step in the third operation (respectively the fourth) needs get the value of block $A_{k,j}$ ($k=1…q$; $j=k+1…q$) (respectively $B_{k,j}$ in which $k=1…q$; $j=1…k-1$) of *k* step. As to the fifth operation, we must know that block $A_{i,j}$ ($i=1…q$ *and* $i\neq k$; $j=k+1…q$) and $A_{i,k}$ ($i=1…q$ *and* $i\neq k$; $k=1…q$) of *k* step before we compute the value $A_{i,j}$ ($i=1…q$ *and* $i\neq k$; $j=k+1…q$) of *k+1* step. The situation in the sixth operation is just like that in the fifth, we can compute the value $B_{i,j}$ ($i=1,…,q$ *and* $i\neq k$; $j=1…k-1$) of step *k+1* after we getting the value $B_{i,j}$ ($i=1…q$ *and* $i\neq k$; $j=1…k-1$) and $A_{i,k}$ ($i=1…q$ *and* $i\neq k$; $k=1…q$) of step *k*. In the operation '7', to get the value $B_{i,k}$ ($i=1…q$ *and* $i\neq q$; $k=1…q$) of *k+1* we must know the value $A_{i,k}$ ($i=1…q$ *and* $i\neq q$; $k=1…q$) of step *k*.

To express the inter-steps data dependence, a data dependent graph will be drawn. First, we define "directed arc" as follows: a directed arc from block "*X*" to block "*Y*" (*X, Y* is the block of matrix) signifies the operation on block "*Y*"

should not be executed before the operations on block "*X*" are finished. Then we can describe the data dependence using directed arc (See Fig.5). 'Loop 1' in the Fig.5 represents the inter-step data dependence of operation '1' and operation '5'. Operation '5' at the step 2 must finish before we execute the operation '1' at the step 3. Fig 4 shows us all the inter-steps data dependence existing between step 2 and step 3 with q is 5.

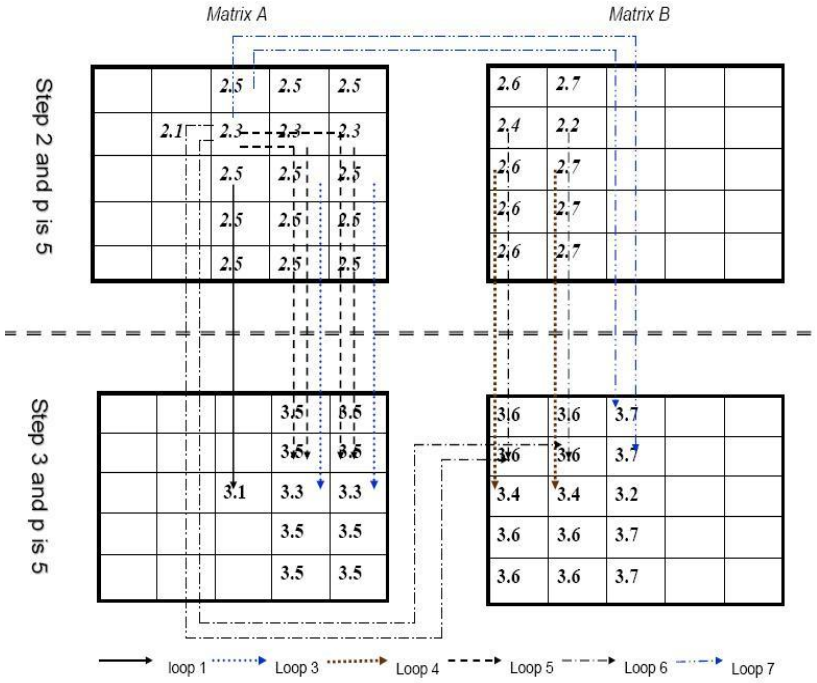


Fig.4. Inter-steps Data Dependence

We have analyzed all the inter-steps data dependence using Fig.4 and the intra-step based data dependence using Fig.3. Then we can summarize all the data dependence using Fig .5.

3. Formal Description of Data Dependence

The data dependence is up to write operation on blocks. It is the data operation that determine when and which blocks can be executed.

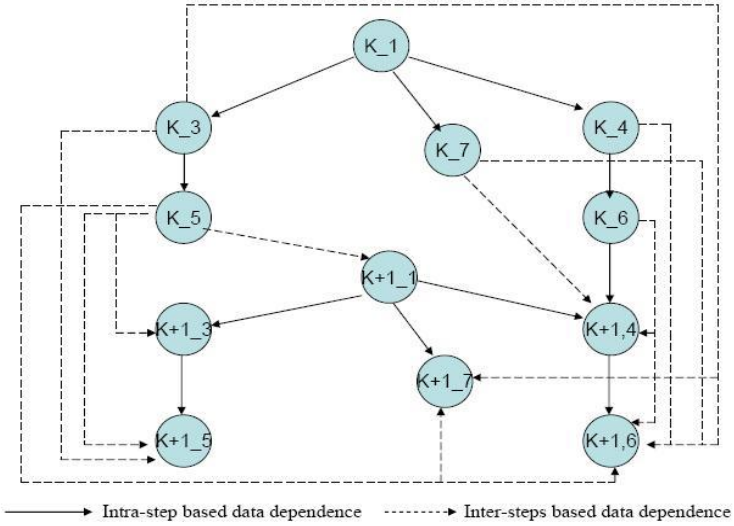


Fig.5. All the Data Dependence in the algorithm. The letter k and k+1 stand for the step k and step k+1 during algorithm execution. The number of behind letter is used to show the operations marked in algorithm 1.

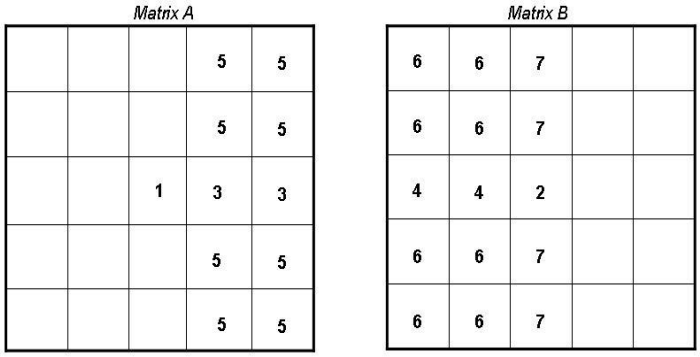


Fig.6. The data operation at step 2 with q is 5 in Algorithm 1

Fig.6 shows us that write operation on each block after executing one iterative step. These operations can be divided into two parts according to its goal written into (matrix *A* or matrix *B*). Operations '1', '3' and '5' are on matrix *A*, while operation '2', '4', '6' and '7' are on matrix *B*.

It is the many similarities existing in write and read operation on matrix A and matrix B both in the intra-step parallelism and in the inter-steps parallelism that we will take matrix A and matrix B together into account as an augmented matrix.

Definition 1: Let
AB(i,j) = A(i,j) i=1…q; j=1…q
AB(i,j+q) = B(i,j) i=1…q; j=1…q
$AB^{k}_{i,j}$ represents the block $AB_{i,j}$ at step k for k=1…q. Let us represent domain $AB^{k}_{i,j}$ using a triplet (k,i,j), then we can use the triplet KIJ = {(k,i,j) | k,i=1…q and j=k+1…k+q} to mark the status of block operation, i.e. KIJ represents the operation made at the step k in the row i and column j of augmented matrix $AB_{i,j}$. For the algorithm, one $AB^{k}_{i,j}$ is modified once and only once per iterative step. So these triplets can be used to describe the sequence of operation on augmented matrix. Consequently we can use the triplet as a global signal to control the data write operation, thus the data dependence in the algorithm can be describe using these triplets.

Definition 2: Let the binary relation $\succ$ be defined as follows: $X \succ Y$ (*X,Y* is the block of matrix) if and only if there exists an edge from *X* to *Y* in the Fig.5 . The data dependence can thus be represented by the triplets defined above. For all the data dependence existing in Fig.5, we have:

Intra-step data dependence can be written as follows:
(k, k, k)$\succ$(k, k, j) *k=1… q; j=k+1… q*
(k, k, k)$\succ$(k, k, j+q) *k=1… q; j=1…k-1*
(k, k, j)$\succ$(k, i, j) k=1…q ; j=k+1…q; i=1…q and i≠k
(k, k, j+q)$\succ$(k, i, j+q) k=1…q; j=1…k-1; i=1…q and i≠k
(k, k, k)$\succ$(k, i, k+q) k=1…q; i=1…q and i≠k

Inter-step data dependence can be summarized as follows：
(k-1, k, k)$\succ$(k, k, k) *k=2…q*
(k-1, k, j)$\succ$(k, k, j) *k=2…q; j= k+1…q*
(k-1, k, j+q)$\succ$(k, k, j+q) *k=2… q; j=1…k-1*
(k-1, i, k)$\succ$(k, i, j) k=2…q; j=k+1…q; i=1…q and i≠k
(k-1, i, j)$\succ$(k,i, j) k=2…q; j= k+1… q; i=1…q and i≠k
(k-1, i, j+q)$\succ$(k, i, j+q) k=2…q; j=1…k-1; i=1… q and i≠k
(k-1, i, k)$\succ$(k, i, j+q) k=2…q; j=1…k-1; i=1… q and i≠k
(k-1, i, k)$\succ$(k, i, k+q) *k=2…q; i=1…q and i≠k*

Now we will summarize binary relation obtained above according to operation marked in algorithm 1.
(k, k, k)$\succ$(k, k, j) k=1… p; j=k+1…p **(1)**
(k, k, k)$\succ$(k, k, j) and (k-1, k, j)$\succ$(k, k, j) k=2…p; j=k+1… p **(3)**
(k, k, k)$\succ$(k, k, j+q) and (k-1,k, j+q)$\succ$(k, k, j+q) k=2… p; j=1… k-1 **(4)**
(k,k,j)$\succ$(k,i,j) and (k-1,i,k)$\succ$(k,i,j) and (k-1,i,j)$\succ$(k,i,j) *k=2…q;j=k+1…q; i=1… q and i≠k* **(5)**
(k,k,j+q)$\succ$(k,i,j+q) and (k-1,i,j+q)$\succ$(k,i,j+q) and (k-1, i, k)$\succ$(k, i, j+q) k=2…q; j=1…k-1; i=1…q and i≠k **(6)**
(k,k,k)$\succ$(k,i,k+q) and (k-1,i,k)$\succ$(k,i,k+q) k=2…p; i=1…p and i≠k **(7)**

4. Design of Hybrid Parallel Programming Paradigm

As suggested by Ian Foster [13], parallel processing on an algorithm needs four distinct stages which are: partitioning, communication, agglomeration and mapping. Next we will design the parallel paradigm of Gauss-Jordan algorithm according to those four steps.

Because large scale matrix inversion is hard to be executed directly, so partitioning is the first thing to do. In this paper, large scale Matrix *A* has been partitioned into n×n blocks, referring to Fig.1 and Fig.2. After partitioning, the objects of all the operations in the algorithm focus on blocks (sub-matrices), which makes a more complex problem decomposed into some easy-to-solve sub-problems. No communication between different small sub-problems exists during the process of program execution. The communication is just between matrix *A* and its sub-matrix. The operation on sub-matrix can read from (or write back to) the corresponding blocks of Matrix *A*. We can design this stage of paradigm according to parallel divide and conquer model in which the sub-problems can be solved at the same time, giving sufficient parallelism.

Analysis in the Section 2 tells us that inter-steps and intra-step parallelism exist in this algorithm and data-dependences on different blocks determine the sequence of different operations on a block (sub-matrix). Fig.1 shows us the inter-step data dependence in Matrix *A* and 12 blocks has the same write-operation (operation '5'). They have the same operation and there is no communication between them. I.e. the sub-tasks of the algorithm are independent and each processor can execute one part of them. So this kind of operations can use single program multi data paradigm model to deal with. As to Fig.4, the operation represented by loop '1' shows us that before the operation '1' in step 3 begins, the operation '5' in step 2 must finish. This kind of parallelism we can use data pipelining paradigm to deal with them.

As to the stages of agglomeration, mapping. Because all the write-operations of the algorithm are based on sub-matrix and the results will be return to the Matrix *A* after any a step of operation, so agglomeration is finished when all the operations on sub-matrix are end. After the hybrid programming paradigm, as much as sub-tasks will be generated and you can use appropriate schedule stratagem to map the sub-tasks to computing resources.

Next, a Flow-chart of hybrid programming paradigm can be seen in Fig.7. Based on the analysis above, a formal description of parallel programming paradigm of BbGJ for desktop grid platform can be presented using Algorithm 2 in this paper.

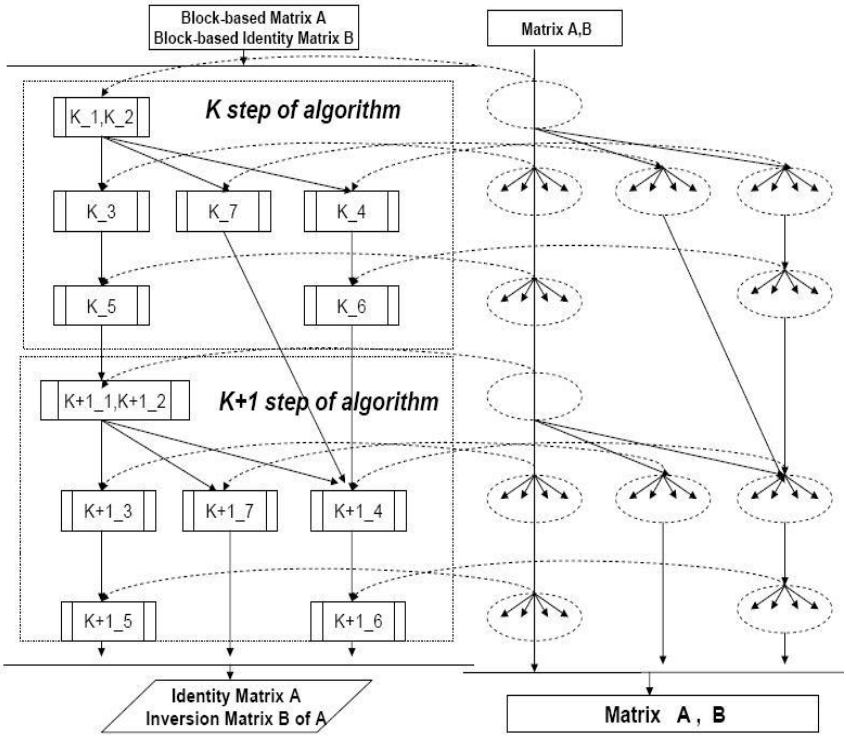


Fig.7. Left part is the flowcharts of the paradigm execution and the right part is the corresponding data operation. q steps needed to execute in the paradigm and the execution between step 'k' and step 'k+1' is described. In the right parts, 'many arrows' represents these data can be executed simultaneously.

The series of three tuples in new hybrid program paradigm for desktop grid system can be described using Fig.8. The Fig.9 shows us the workflow of different operations in the algorithm. The execution can be described using a series of three tuples. The key path of those three tuples is the base of setting the propriety of different operation. Three-level based propriety is set in the hybrid algorithm for desktop grid system.

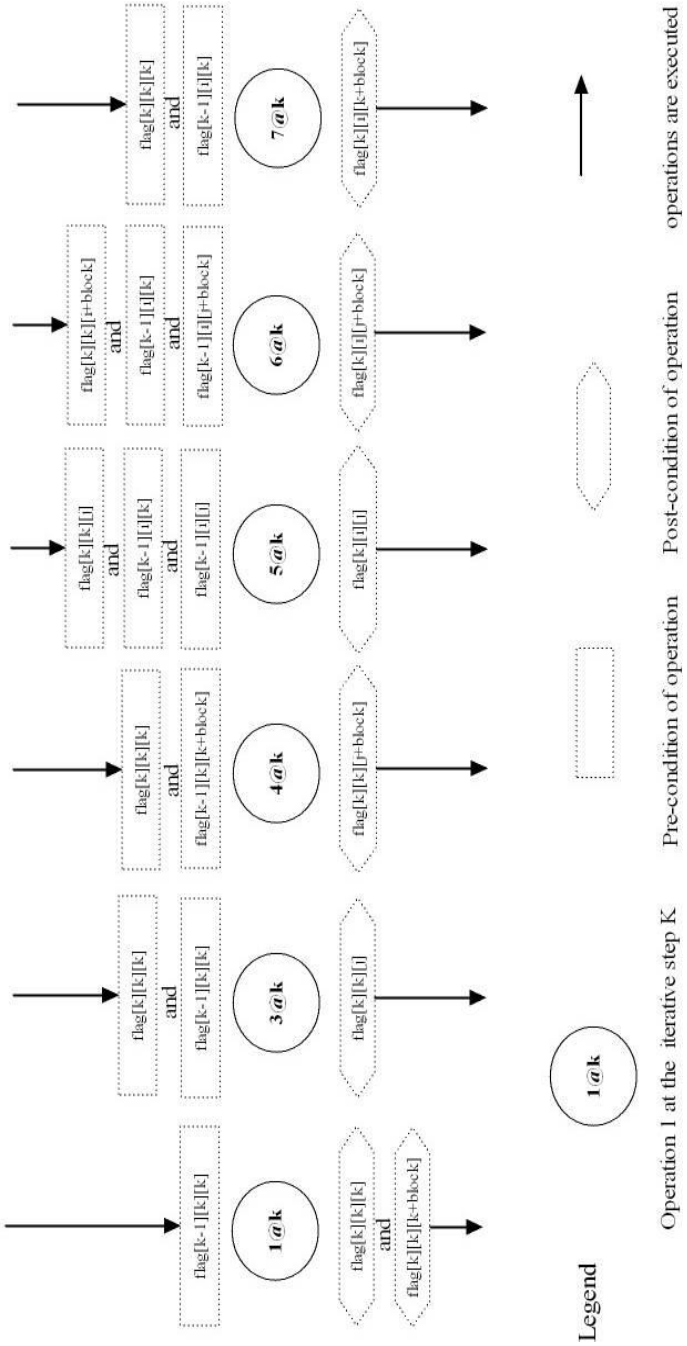


Fig.8. The series of three tuples of BbGJ in desktop grid system

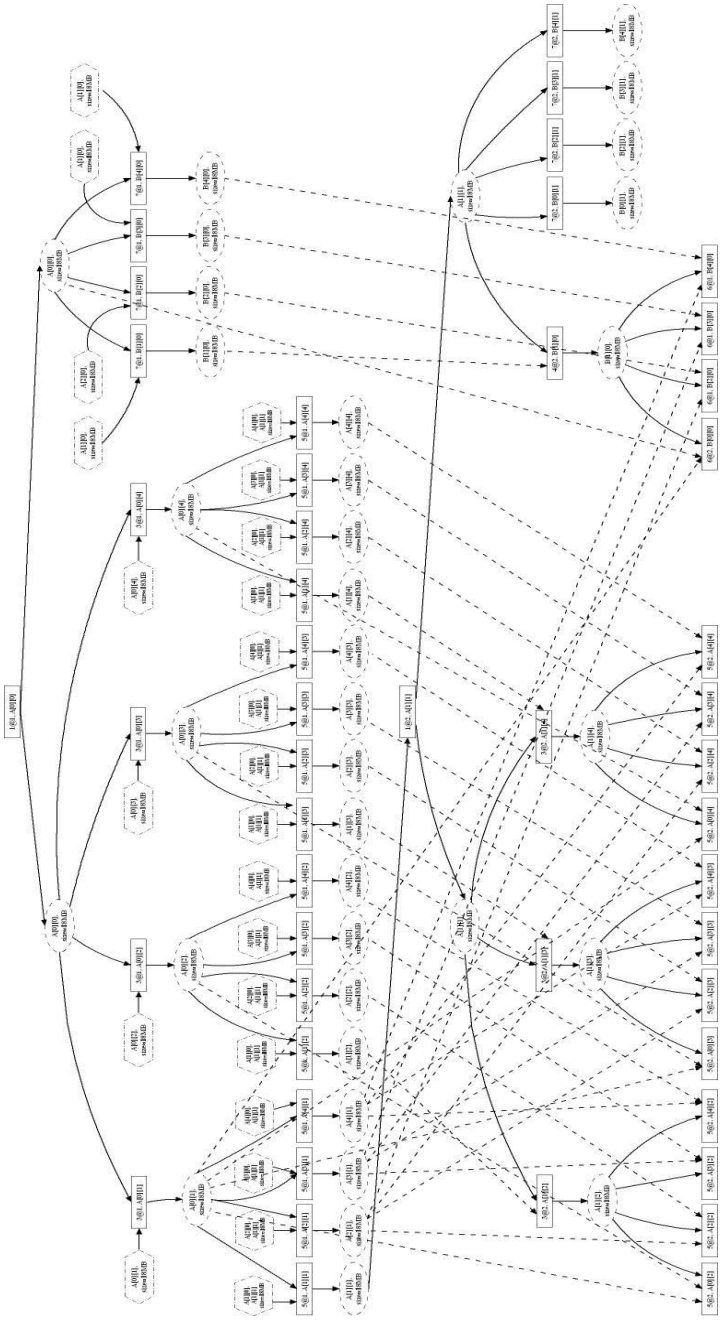


Fig.9. Workflow of three tupes in the BbGJ

Hybrid parallel algorithm for desktop grid system

Input: $A,B \leftarrow I(n,n)$, q
the logical value of (0,1,1) is set to be true;
for(i=1;i<=q;i++) the logical value of (0,1,i) is set to be true;
for(i=1;i<=q;i++) the logical value of (0,i,1) is set to be true;
for(i=1;i<=q;i++)
for(j=1;j<=q;j++)
the logical value of (0,i,j) is set to be true;
Output: $B=A^{-1}$

---

if the logical value of (k, k, k) is true
then $A^{k}_{k,k} \leftarrow (A^{k-1}_{k,k})^{-1}$;
the priority of this operation is set 1(highest level)
the logical value of (k, k, j) is set to be true;
end if
//
if the logical value of both (k, k, k) and (k-1,k,j) are true
then $A^{k}_{k,j} \leftarrow A^{k}_{k,k} A^{k-1}_{k,j}$ (k=2,...,q; j=k+1,...,q)
the priority of this operation is set 1(highest level)
the logical value of (k, k, j) is set to be true;
end if
//
if the logical value of both (k, k, k) and (k-1,k,j+q) are true
then $B^{k}_{k,j} \leftarrow A^{k}_{k,k} B^{k-1}_{k,j}$ (k=2,...,q; j=1,...,k-1);
the priority of this operation is set 2 (important level)
the logical value of (k, k, j+q) is set to be true;
end if
//
if the logical value of (k, k, j) and (k-1,i,k) and (k-1,i,j) all are true
then $A^{k}_{i,j} \leftarrow A^{k-1}_{i,j} - A^{k-1}_{i,k} A^{k}_{k,j}$ (k=2,...,q; j=k+1,...,q; i=1,...,q and $i \neq k$)
the priority of this operation is set 1 (highest level)
the logical value of (k, i, j) is set to be true;
end if
//
if the logical values of (k, k, j+q) and (k-1,i,j+q) and (k-1,i,k) all are true
then $B^{k}_{i,j} \leftarrow B^{k-1}_{i,j} - A^{k-1}_{i,k} B^{k}_{k,j}$ (k=2,...,q; j=1,...,k-1; i=1,...,q and $i \neq k$)
the priority of this operation is set 2 (important level)
the logical value of (k, i, j+q) is set to be true.
end if
//
if the logical value of both (k, k, k) and (k-1,i,k) are true
then $B^{k}_{i,k} \leftarrow -A^{k-1}_{i,k} A^{k}_{k,k}$ (k=2,...,q; i=1,...,q and $i \neq k$)
the priority of this operation is set 3 (normal level)
the logical value of (k, i, k+q) is set to be true.
end if

---

5. Performance Evaluation

In this section, we first evaluate the performance of traditional BbGJ algorithm in real desktop grid environment. The second experiment is to show the performance of our adapted algorithm in real desktop grid environment. Then, we compare the performance of our algorithm and traditional one in ideal desktop environment. All this experiments are based on YML/XtremWeb [5][6][9] (middleware) on desktop grid platform through the comparison the hybrid paradigm with intra-step parallel programming paradigm[3][7][15]. To state conveniently, we call the hybrid parallel algorithm proposed in this paper for desktop grid environments as "BbGJ_DG". Here what we want to emphasize is that all the experiments are just to show the performance of hybrid programming paradigm, no special middleware is necessary, you can choose any middleware what you like. In our experiment, the topology of computing resources can be described using Fig.10 and the computational resources can be described as follows:

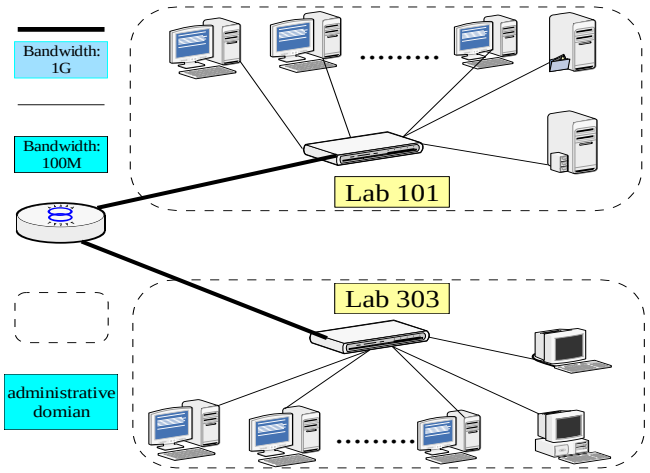


Fig.10. The topology of desktop grid platform

Table 1. Resources in Desktop Grid platform

Site	Node	CPU/Memory
Lab 303	16	Inter 2.66GHz/512M
Lab 101	64	AMD 1.8 GHz/512M

5.1. The character of desktop grid environment

Experiment motivation: test the performance of traditional algorithm in the desktop grid environment.

Experiment Environment: the first case is 30 nodes based desktop grid platform in Lab 101. The experiment time is set in the morning when the computing resources are high volatile. The second case is based on 30 nodes from Lab 101 (20 nodes) and Lab 303 (10 nodes)

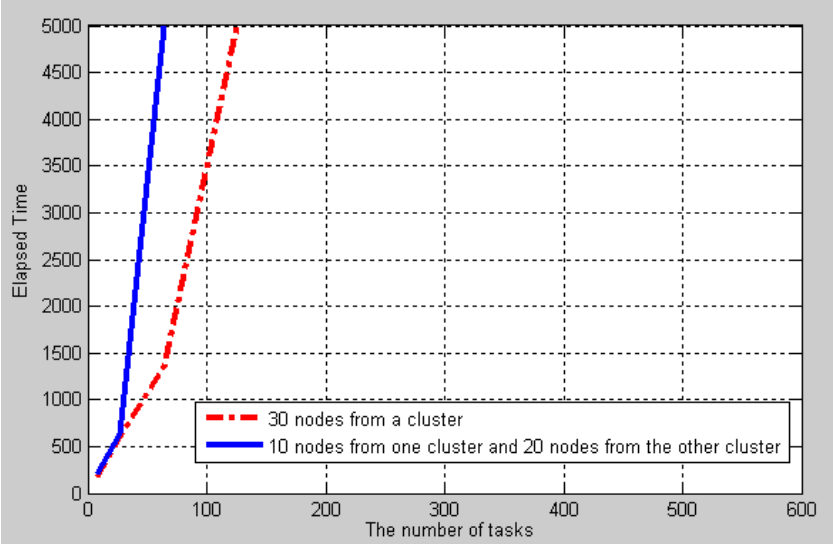


Fig.11. The performance of traditional version in desktop grid environment

As we mentioned above, the key character of desktop grid environment is its volatility [11,17] which causes frequent task migration between computing nodes. Paper [16] also emulate BbGJ algorithm in the desktop grid environment and the conclusion is that perhaps 163 days can finish the whole program. This experiment is also to evaluate the performance of data dependence based algorithm in desktop grid environment. In this experiment, we suppose when the execution time is more than 5000 (s), we think the program can't be executed successfully, i.e. the program can be finished in this environment.

Fig.11 shows us the when the number of tasks is not large, the program can be finish and when the number becomes larger, the program can't be executed successfully. From this experiment, we can conclude that when there is data dependence between operations in the parallel algorithm, we can't get better performance using directly adapted version of parallel algorithm. It is better for us to take volatility into consideration and multi-copies of the task have to be sent to more computing nodes. Also, the more important tasks which is in the key path have the priority to be executed. Only

those factors are considered during allocating tasks to computing nodes, the better performance can be achieved. The next experiments just to show the new hybrid parallel algorithm proposed in the paper which takes those factors presented above can get better performance.

5.2. The performance of "BbGJ_DG" in desktop grid environment.

Experiment motivation: test the performance of "BbGJ_DG" in desktop grid environment.

Experiment Environment: the first case is 30 nodes based desktop grid platform in Lab 101. The experiment time is set in the morning when the computing resources are high volatile. The second case is based on 30 nodes from Lab 101 (20 nodes) and Lab 303 (10 nodes)

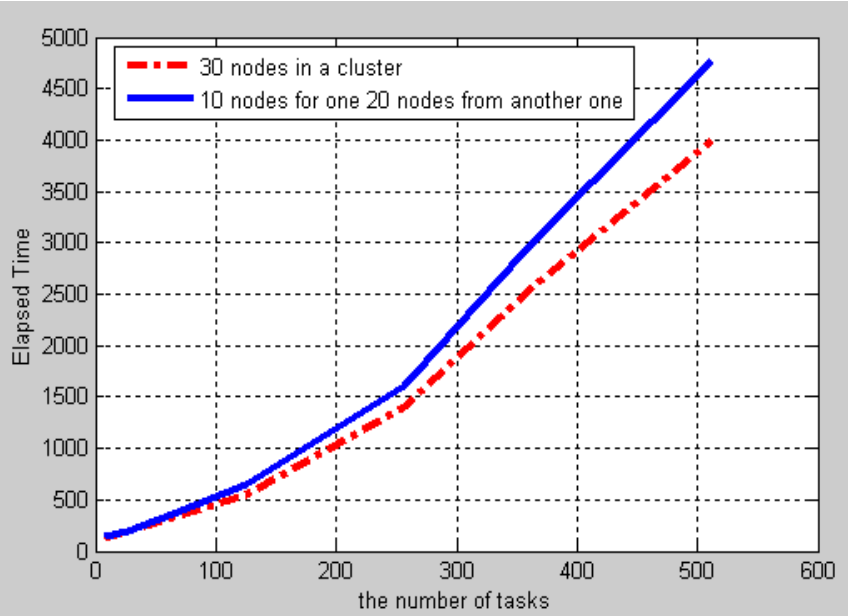


Fig.12. The performance of "BbGJ_DG" in desktop grid environment

The experiment in section 5.1 show that traditional version of BbGJ can't adapt to desktop grid environment and in more case, the program can't be executed successfully. In this paper, a new parallel adapted version which is called as "BbGJ_DG" takes the character of desktop grid environment into consideration. The experiments testify that, BbGJ_DG can be executed successfully on time. The program can be finished in two cases. So the conclusion can be achieved that BbGJ_DG can adapt to desktop grid environment. The difference between two cases in this experiment is from the

experiment environments, which is cluster based desktop grid environment and while the other is grid based desktop grid environment.

5.3. Time difference between BbGJ_DG and traditional one

Experiment motivation: test the performance of BbGJ_DG with different granularity. Two methods to change the granularity of parallelism in this experiment are changing block-size with block-count fixed and changing the block-count with block-size fixed.

Experiment Environment: 45 nodes in Lab 101 and 15 nodes in lab 303. The experiment time is set at midnight (begin at 0 o'clock) at which all the computing resources can be seen as dedicated computing resources. This environment can be seed as ideal environment.

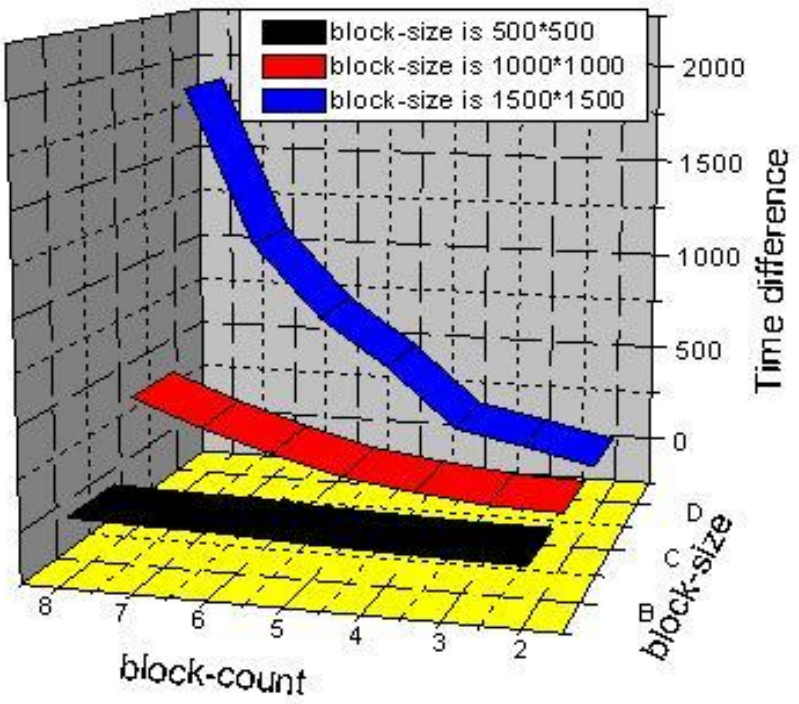


Fig.13. Time difference becomes very large when the block-size of matrix increase

From the Fig.13, we can find that when the block-size is not large, the elapsed time of algorithm based on BbGJ is just a little less than that based on intra-step paradigm (traditional parallel algorithm). And time gap becomes larger with the increase of block-size. The reason is that when block-size is small, more time is consumed in communication and schedule while time

used to compute is small. So the advantage of BbGJ_DG is not obvious. From the analysis in section 3 and 4, we know the advantage of BbGJ_DG is to make full use of computing resources in desktop grid platform, i.e. the BbGJ_DG can make all the tasks available to be executed to be executed in parallel. With the increase of block-size, the advantage of BbGJ_DG becomes more and more obvious. the similar conclusion can be achieved when changing block-count and fixing the block-size

Time difference becomes larger with the condition of block-count change. With the increase of block-size, the computing time takes a great proportion and better concurrency in the BbGJ-DG ensures maximizing parallelism of program execution, which shortens the wait time comparing with intra-step paradigm. The same reason can explain the gap of time difference becomes larger with the increase of block-count. In the program based on BbGJ_DG, less wait time caused by data dependence is consumed in the process of program execution. Because hybrid paradigm make all the possible parallelizable parts executed in parallel and intra-step based paradigm just ensure the parallelism in the loop without considering the inter-step parallelism.

6. Conclusion

This paper presents a hybrid parallel adapted version of BbGJ for desktop grid environment based on the analysis of data dependence in the algorithm. The new paradigm, which exploits all the possible parallelizable parts of the algorithm, can help us improve the performance of block-based Gauss-Jordan algorithm on the desktop Grid platform in which more computing resources can be harnessed. At the same time, BbGJ_DG take the character of desktop grid system into consideration. So BbGJ_DG can get better performance in desktop grid environment. The experiment also testify the good performance when solving a real problem, and its scalability makes sure the paradigm can be tailored to more computing resources in desktop grid environment very easily.

Acknowledgements. This paper is supported by the Open Foundation of State Key Laboratory of Hydrology-Water Resources and Hydraulic Engineering (Grant No. 2011490804).

References

1 S. Petiton: Parallelization on an MIMD computer with realtime Scheduler. Aspects of Computation on Asynchronous Parallel Processors, North Holland, (1989).
2 N. Melab, E.-G. Talbi, and S. G. Petiton: A parallel adaptive Gauss-Jordan algorithm. The Journal of Supercomputing, 17(2):167–185, (2000).

3 M. Hugues. Algorithmique scientifique distribu grande chelle et programmation yml sur cluster de clusters. mastersthesis, (2007).
4 N. Melab, E.-G. Talbi, and S. G. Petiton: A parallel adaptive version of the block-based gauss-jordan algorithm. In IPPS/SPDP, pages 350–354, (1999).
5 O. Delannoy and S. Petiton: A Peer to Peer Computing Framework: Design and Performance Evaluation of YML. In Third International Workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Networks, pages 362– 369. IEEE Computer Society Press, (2004).
6 O. Delannoy, N. Emad, and S. G. Petiton. Workflow Global Computing with YML. In The 7th IEEE/ACM International Conference on Grid Computing, pages 25–32, (2006).
7 L. M. Aouad and S. G. Petiton. Parallel basic matrix algebra on the grid'5000 large scale distributed platform. In CLUSTER, (2006).
8 F. Cappello et al: Grid'5000: a large scale and highly reconfigurable grid experimental testbed. In The 6th IEEE/ACM International Conference on Grid Computing, pages 99–106, (2005).
9 F. Cappello, S. Djilali, G. Fedak, T. Herault, F. Magniette,V. N′ eri, and O. Lodygensky. Computing on LargeScale Distributed Systems: Xtremweb Architecture, Programming Models, Security, Tests and Convergence with Grid. Future Generation Computer Science, (2004).
10 S. T. Ian Foster, C. Kesselman. The Grid: Blueprint for a Future Computing Infrastructure. Morgan Kaufmann, USA, (1999).
11 Ling Shang, Zhijian Wang, Xiaofeng Zhou, Xiaoping Huang, Yongshang Cheng: TM-DG: a trust model based on computer users' daily behavior for desktop grid platform . ACM and ACM SIGPLAN Proceedings of the 2007 symposium on Component and framework technology in high-performance and scientific computing, Montreal, Quebec, Canada (2007)
12 Miquel Pericàs, Ruben Gonzalez, Francisco J. Cazorla, Adrian Cristal, Daniel A. Jimenez and Mateo Valero.: A Flexible Heterogeneous Multi-Core Architecture. Parallel Architecture and Compiler Techniques (PACT-2007)
13 Designing Parallel Algorithms : http://www-unix.mcs.anl.gov/dbpp/text/book.html
14 Parallel Programming Models and paradigms: http://www.buyya.com/cluster/v2chap1.pdf
15 Maxime Hugues and Serge G. Petiton: A Matrix Inversion Method with YML/OmniRPC on a Large Scale Platform. VECPAR'2008, Jun 24-27, 2008, Toulouse, France (2008).
16 S. Petiton and L. Aouad. Large scale peer to peer performance evaluations, with Gauss-Jordan method as an example. Proceedings of the PPAM2003, fifth international conference on parallel processing and applied mathematics. Poland, Sep. 2003. Lecture Notes in Computer Science 3019, 938-945, (2004).
17 Derrick Kondo, Gilles Fedak, Franck Cappello, Andrew A. Chien, and Henri Casanova. 2006. On Resource Volatility in Enterprise Desktop Grids. In Proceedings of the Second IEEE International Conference on e-Science and Grid Computing (E-SCIENCE '06). IEEE Computer Society, Washington, DC, USA, 78- (2006)
18 Derrick Kondo Eric Wing. 2002. Models and Scheduling Mechanisms for Global Computing Applications. In Proceedings of the 16th International Symposium on Parallel and Distributed Processing (IPDPS '02). IEEE Computer Society, Washington, DC, USA, (2002)
19 D.P. Anderson and G. Fedak. The Computational and Storage Potential of Volunteer Computing, IEEE/ACM International Symposium on Cluster Computing and the Grid, Singapore, May 16-19, (2006).

Yizi Shang is a research assistant of hydro-informatics at Tsinghua University. He received B.Sc. degree in computers and electrical engineering from North China University of Water Resources and Electric Power in 2005, and the M.S and Ph.D degree in hydro-informatics from Tsinghua University in 2010. He was a research fellow of school of engineering and applied sciences (SEAS) at Harvard University from 2008 to 2009. His research interests concentrate on the application of information and communications technologies in addressing the increasingly serious problems of the equitable and efficient use of water for many different purposes.

Guiming LU is a professor at school of information engineering in North China Universtiy of water conservancy and electric power. His research and interests cover distributed computing, parallel computing, Grid computing.

Ling SHANG is an associate professor at school of information engineering in North China Universtiy of water conservancy and electric power. His research and development interests cover distributed computing, parallel computing, Grid computing and Cloud computing.

Guangqian Wang is a professor of Department of Hydraulic Engineering at Tsinghua University. His areas of expertise include hydro-informatics and computation fluid. He has published more than 100 technical papers and authored or coauthored several books in his areas of expertise. Dr. Wang is also the director of the State Key Laboratory of Hydro-science and Engineering, Tsinghua University.

Received: September 07, 2010; Accepted: January 16, 2011.

DOI: 10.2298/CSIS100915027F

Problem Solving by soaking the concept network

Xixu Fu and Hui Wei

College of Computer Science, Fudan University
ShangHai, China
081024009@fudan.edu.cn

Abstract. Because of the complexity and fuzziness of the real world, it's hard to build a dense knowledge system and reason in it with traditional methods. But man can deal with such tasks freely. Inspired by cognition and linguistics, a solution is advanced for reasoning dense knowledge in this paper. Objects and concepts are organized in the form of concept graph. Soaking the nodes in the graph until the result is represented in the graph the final graph can be the explanation of the scenario. With the naïve algorithm, monotonic scenario reasoning problem can be solved in dense knowledge environment.

Keywords: cognition, scenario reasoning, knowledge representation, object-oriented, concept graph, soak.

1. Introduction

Natural world is a complex world. P. J. Hayses defined a good knowledge system to be a dense knowledge system with detailed description for objects [27]. However, it is hard to explain how a scenario taken place for machines.

There are two main problems in the construction of expert systems [30]: the bottleneck of knowledge acquisition and the narrow scope of knowledge system. The former involve that how to translate the knowledge in the real world into the knowledge that the expert system can use, while the latter involve that expert system only adapt to domain-restricted problem solving, once related to the issue of domain-opened, its capacity of problem solving will become very vulnerable. The main reason for these problems is that the knowledge of expert system is almost mechanical and lack of the support of the underlying semantics. They are not from the perspective of the development of concept to construct a concept system.

Association is a main reasoning method of human. It's a powerful method in reasoning. For dense knowledge system, association can be an efficient method. Let's see the example below.

Once, a little horse went across a little river with a bag of salt on its back. It bumped into water by mistake. But after the mistake, it found the bag become light and feel happy. The next day the little horse went cross the river with a bag of cotton on its back. It felt into the water on purpose. But this time it felt heavier after stood up.

Men use association to solve these problems while computers search state space for such tasks. We can naturally understand the story. But for computer algorithms, it's hard to define and get the result. There are even many muddy tasks without precise definition. For these problems, traditional state-space searching can do nothing.

Relations are complex in actual problems. Connectionism [7] is suitable for complex problems, but connectionism doesn't maintain accurate relations. This may cause errors in the reasoning process.

In developmental psychology, objects are always the focus of children in knowledge acquisition. The thinking process also starts from object. When all the details loaded into the mind, the process is built.

Inspired by psychology and linguistics, a method is advanced to simulate man's association and solve many actual problems. Objects and concepts are regarded as basic elements. Begin with known objects, the method soak the knowledge network and activate more and more concepts and objects. When all the nodes are related, the problem is solved.

In this paper, related works are first analyzed first. Then, basic elements and concept network is defined. After that, dynamic soaking process is analyzed with an example. The algorithm is provided next based on the analysis. At last, some discussions are carried about efficiency and future works of the method.

2. Related Work & Motivation

2.1. Symbolism and connectionism

The advocates of symbolism try to represent knowledge with symbols. Symbols can form reasoning precisely, but it is hard to build a complex system with symbols. Connectionism [7] is suitable for complex problems, but connectionism doesn't maintain accurate relations. This may cause errors in the reasoning process. The problem is to find basic elements that can both represent complex semantics and relation.

2.2. Linguistic Semantics & Pragmatics

Language reflects the method man's comprehension of the world. Words are potential elements describing the world. They can be important reference in building knowledge systems. Words and grammar build the semantic of language [12].

Real nouns [11] are basic elements of language and cognition. Children first recognize them. Verbs and rhetoric words such as adjectives and adverbs are acquired consequently. Reference words such as prepositions

and pronouns are aware last [13]. These are basic element of human knowledge. Because of homonymy, polysemy, semantic motivation and the changing of background linguistic Semantics and Pragmatics varies a lot. Language cannot be a unified representation and reasoning tool for machines.

2.3. Linguistic Based Knowledge Systems

There are many knowledge systems based on linguistic semantics.

CYC [19] is a well known linguistic based knowledge system. It advanced the idea to build a large-scale knowledge system that can solve various artificial intelligent problems. The system maintains a large knowledgebase with common sense. With the reference of common sense, the system can solve many common sense concerned problems such as NLP.

WordNet [18] is a well-known semantic dictionary. It organizes words by their semantic and lists semantically related words and syntax. Many tasks have been carried out with this system.

These linguistic based system solved many problems about semantics and syntactic. But process concerned reasoning cannot be implemented based on these systems. Semantics of these systems are not abundant to solve complicated problems.

2.4. Semantic Network

Semantic network [14, 17] is a popular knowledge representation. Many knowledge-based applications [15, 22] are built based on knowledge in this format. Objects are related with slots named isa and ako. The semantic of semantic network is not rich enough to support process reasoning.

2.5. Searching based Methods

Searching based methods have been widely used in simple problem solving tasks. These methods include state-space searching and encoded searching methods such as evolutionary computing and genetic algorithm.

State-space searching [3] algorithms can find a way to solve some model problems in which all states are known. But state-space searching is time consuming and limited by knowledge structure [1]. For complex problems, it's almost impossible to find what happened.

State-space searching has its shortcomings. According to Alexander's paper [1], five problems are concerned:

- State Space Search has excessive space requirements.
- State Space Search is time-consuming.
- State Space Search is difficult to implement.
- State Space Search can only be used in uniform trees.

– State Space Search dominates directional searches.

Further more, states are not easy to list and potential states are of great number. Although many heuristic algorithms such as A* [3,4] are invented to reduce the complexity. They can't be applied to complex process reasoning tasks.

Evolutionary computing and genetic algorithm [29] belong to another type of searching algorithms. State-space is encoded into bits or characters. Searching is carried out by intersection and mutation starting from random individuals.

2.6. Reasoning & Association

Fortunately, man's knowledge acquisition and thought can be important reference of building knowledge systems. Association is an important ability of human. Some works related to human association have been advanced in field of neural network and pattern recognition [28].

Children recognized objects first, and then acquire the features of recognized objects. They relate objects together by their reactions and similarity. When thinking, we focus on objects first too. For example, when we see a fly and a magazine, we first focus on fly and magazine, then know we hate the fly and we can kill it with tool such as a magazine, after that we kill the fly with the magazine.

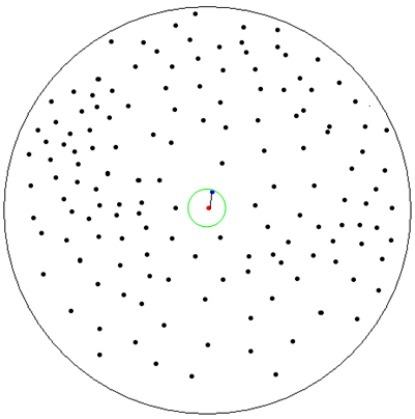


Fig. 1. shows a task that to find nearest point to the red point in a large space. There are large amount of points in the whole space, but most of them needn't be concerned in humans' opinion although computers trends to search all or many of them.

As shown in figure 1, traditional computer algorithms focus on the whole state space while men focus on related concepts only. Although many heuristic algorithms have been advanced to limit the searching space, traditional methods can't tell how a natural scenario takes place. Men's

association process can provide a way for reasoning in complicated knowledge environment.

2.7. Previous Work and Basic Theory

Some work has been done before this work. Objects, concepts and scenarios and related operations are defined in Xixu Fu and Hui Wei's previous work about knowledge architecture [20]. Michael Freund's work [16] and John F. Sowa's book [17] provide basic theory about knowledge representation and identification.

3. Concept Network

The concept network is the core of the problem solving system. Problems are resolved into objects which relate each other with concepts and operations. Objects activate each other according to the relations, and the activating process constructs the whole process of the problem.

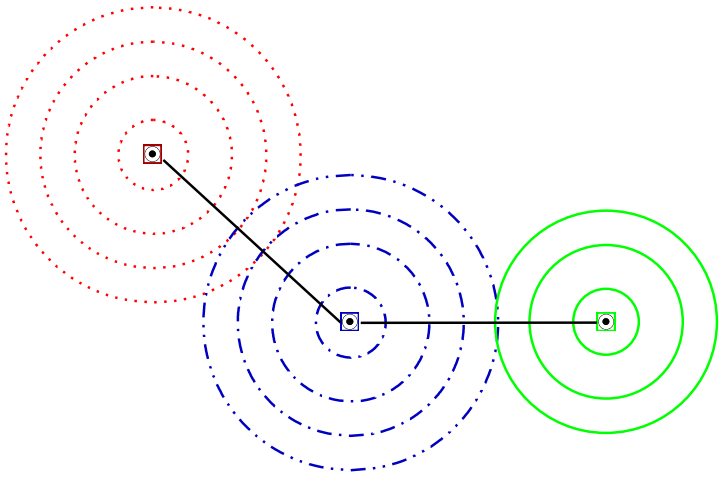


Fig. 2. This figure shows the process of association. Soaking begin with known nodes which are represented by points in the figure. Related nodes are activated to connect the initial nodes together.

3.1. Definition of Elements in Concept Network

As an object-oriented system, objects and concepts are the basic elements. Anything are regarded as objects and abstracted into concepts. All the objects

consists the knowledge network. Objects and concepts are regarded as nodes of the concept network. Nodes and relations consists the whole network.

3.1.1. Various types of Objects

There are many types of words including nouns, verbs, and adjectives and so on. Nouns, verbs and adjectives and adverbs are regarded as lexical words. Other are mainly regarded as grammatical words [25]. Lexical words are semantically important. Various kinds of objects exist in the world too. Similar to natural languages, we can categories these objects into four classes:

- Nominal objects are independent objects that can be represented with nouns.
- Actions are objects to describe the behavior of nominal objects.
- Rhetoric objects are objects used to describe features of nominal objects and actions.
- References are objects that indicate the temporal or spatial situation of objects and actions.

Nouns can represent nominal objects whether concrete or abstract. For example, desk and thought are both nominal objects. Nominal objects can have other nominal objects as parts. A proposition or scenario can be regarded as a nominal object.

An action often has an actor and a target. Some acts may have two targets, one is direct and the other is indirect. An action can be represented as a pentad ACTION<Actor, DirectTarget, [IndirectTarget], Process, Result>. Actor is the object that carries out the action. DirectTarget is the object receives the act. IndirectTarget is the media of the act. Result is the result of the action. Process is the process of acting.

Rhetoric objects are rhetoric to nominal objects or actions. Adjectives and adverbs can be regarded as attributes.

A reference is a dyad REFERENCE<Precedent, Hind>. For example, in the proposition A is on B. A is the precedent of on while B is the hind.

Every kind of objects has its distinctive attributes. For the sake of reasoning, every object has an attribute to describe whether it is activated. An activate operation is used to determine whether related objects can be activated and set them activate.

From the analysis of objects, we can define object as below:

Definition 1: Object is a dyad O<D,E>. D is the description of the object which includes the name of objects and components described by private or protected elements. E is the set of components can be described by public elements.

Variables and functions are regarded as objects. For functions parameters are regarded as public elements.

Figure 3 shows the representation of four kinds of objects.

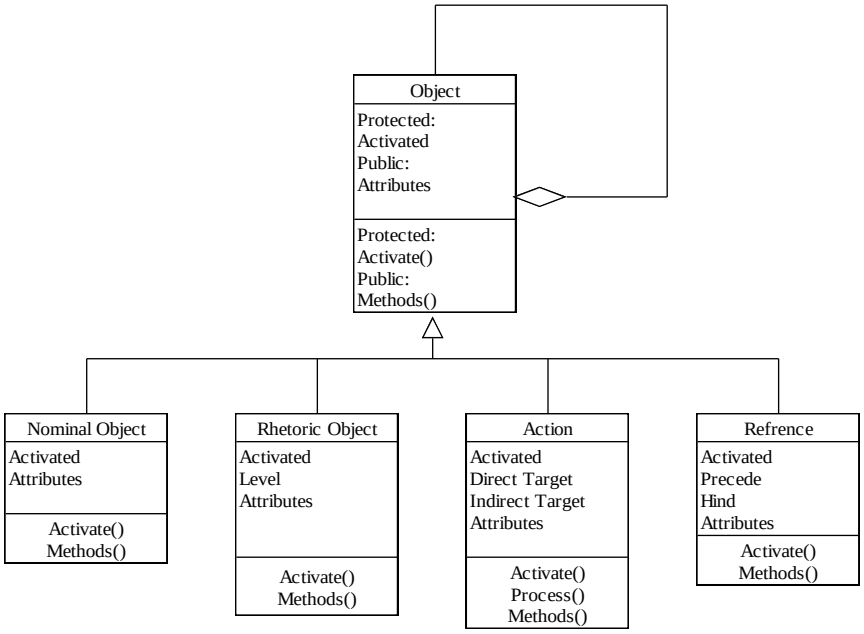


Fig. 3. The figure organized representations of objects and four kinds of objects. Attributes of an object can be regarded as object too.

As shown in Fig. 3, all objects need an activation method to construct the activating process. This method can be included in the constructor of object.

3.1.2. Concepts

Concepts are abstracted from objects. The set of concepts can be represented by a dyad C<D,E> too. It's defined as below:

Definition 2: Concept is a dyad C<D,E>. D is the description of the object which includes the name of objects and components described by private or protected elements. E is the set of components can be described by public elements.

3.1.3. Nodes

Objects and concepts are nodes of concept network. They can have a unified definition as below:

Definition 3: Node is a dyad N<D,E>. D is the description of the object which includes the name of objects and components described by private or protected elements. E is the set of components can be described by public elements.

3.1.4. Relations and Scenarios

Objects and concepts are related with three types of relations: inheriting, convergence and forth putting. When an object or a concept is activated by another object or concept, they become related and a relation is build between them. A scenario is build up with objects, concepts and relations. A scenario can represent a status or a process.

Scenarios are defined to describe things. A scenario is defined as the collection of concepts, objects and their relations. It's defined as below:

Definition 4: Scenario is a triad S<C, O, R>. C is the set of concepts in the scenario. O is the set of objects in the scenario. R is the set of relations in the scenario.

3.1.5. Concept Network

A scenario can be described by concept network which described by nodes:

Definition 5: Concept network is a dyad NW<N, R>. N is the set of nodes in the concept network. R is the set of relations in the scenario.

Concept network can be regarded as a structured node.

3.2. Knowledge Representation with Concept Network

A problem can be represented by a graph consists of concepts and objects as nodes. Nodes are connected with relations. When nodes are activated, they can represent definite fact.

For example, the statement Salt can be dissolved in water can be represented by figure 4.

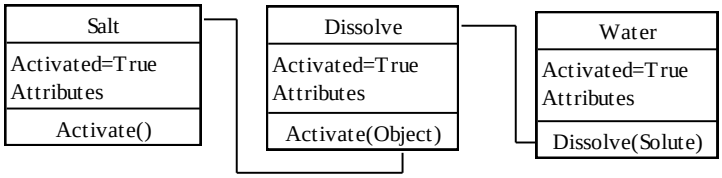


Fig. 4. The Representation of the Statement Salt can be dissolved into water

When all three nodes are activated, the graph means that water can dissolve salt.

3.3. Handle Negations

A graph plus a negative node has the graph as a sub graph. This may course the error that A is not B implies A is B. Negative words such as *no* and *not* are

not regarded as objects or concepts. All objects and are positive objects. Negations should be merged into the node it charges. Conflict should be avoided. For example *not important* may be represented by the object unimportant or not_important. Anything can't be both important and unimportant.

Definition 6: A concept network NC_1 is a sub network of network NC_2 when and only when all the relations and nodes in NC_1 are also in NC_2.

Theorem 1: A concept network implies any of its sub networks.

4. Graph Reasoning

4.1. Overview of the Concept Network Reasoning

4.1.1. Activation of Nodes

A node is activated when the definition has been fulfilled or when it's related to activated objects. There are two major relations between objects. One is inheriting the other is convergence. The activation of nodes can be divided into static activate and dynamic activate.

Definition 7: Static activate is defined as the activation by node structure. Such as a node activated by its components or parent nodes.

Definition 8: Dynamic activate is defined as the activation that a node activate other nodes by its operations.

The activation of nodes can be divided into possible activation and definitely activation.

Definition 9: A node is definite activated if the node can be activated without decision by its activate operation.

Definition 10: A node is possible activated if the node can't be activated without decision by its activate operation.

The activation of nodes can be described below:

- Nodes mentioned as input are regarded as activated objects or concepts.
- If a node is activated then its parent concepts can be activated definitely.
- If a node is activated then objects or concepts as parts of it can be activated definitely.
- A node can be activated definitely by fulfilling the definition in the activate operation.
- An object or concept can be possibly activated when it inherits from an activated concept.
- An object or concept can be possibly activated when some activated objects or concept composes it.

Definite activation can be used in normal reasoning. Possible activation is not used in definite reasoning and only being useful in some possibility reasoning such as imagination. When an object is activated, its attribute activated is set true.

4.1.2. Static View of Problem Solution

Any problem has a start scenario and a result scenario. All nodes in both scenarios and relations in the start scenario are extracted to build a concept network. Then the soaking process begins. When the result scenario can be represented in the concept network, the problem is solved.

4.1.3. Activation Process

An object can activate and relate objects or concepts as its parameters. When a node is activated, an edge between it and the node activate it is added into the reasoning graph. Let's take the problem represented by figure 4 as an example. Let salt in water be input scenario. Figure 5 presents the process of reasoning. The expected result is salt dissolve in water.

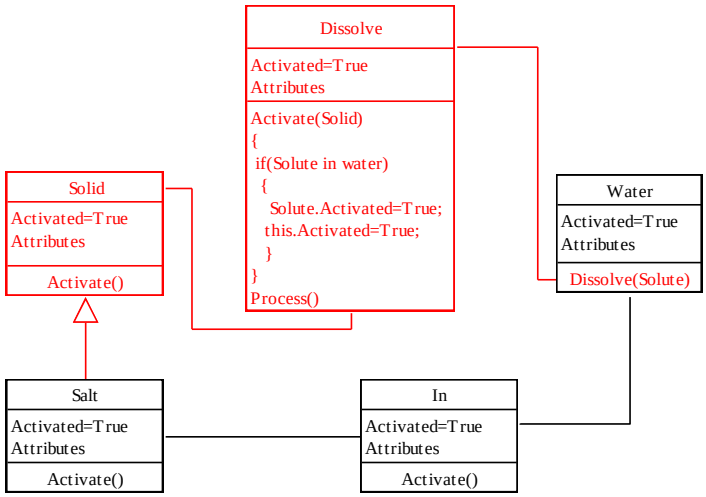


Fig. 5. The Solution of the Problem Salt in Water

The objects *water*, *in* and *salt* are connected in a known scenario. Object dissolve is activated as an operation of water. The concept solid is activated as a parent concept of salt. Then dissolve activates solid and the problem is solved.

4.2. Explain Complex Scenarios by Soaking Concept Network

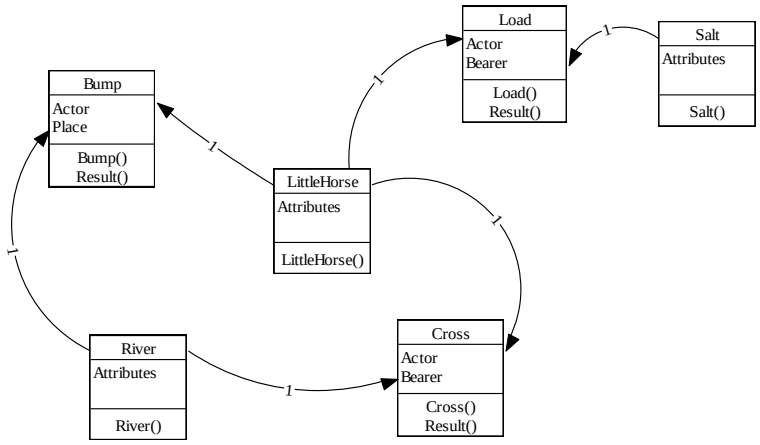


Fig. 6. Initial status of the story can be shown in this figure. The little horse is crossing a river, loading some salt. Arrows mean activation.

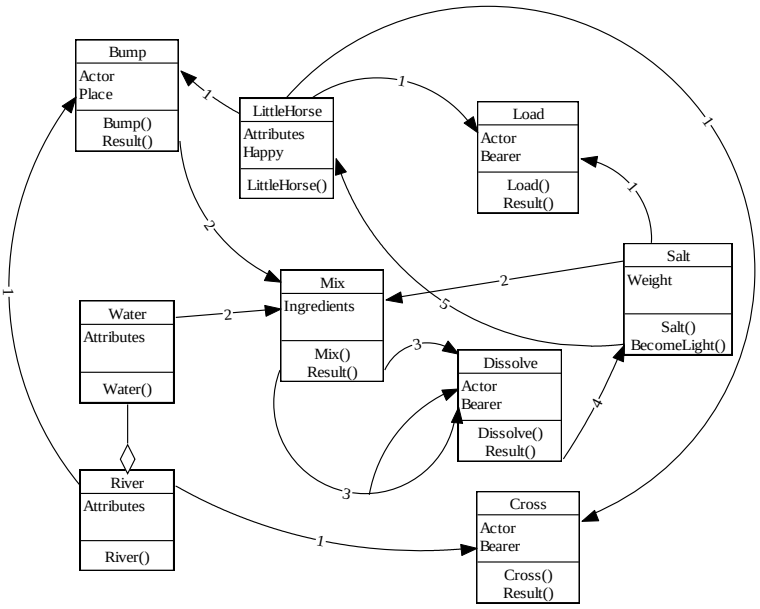


Fig. 7. This figure shows the overall activation process of related concepts. The graph can explain the process of the first scenario. The little horse bumped into river when crossing it. Salt mixed with water and dissolved in the water. So salt become light and the little horse felt happy.

Let's take the story mentioned in the introduction session as an example for scenario comprehension. The story can be represented into two scenarios. The first describes the first time the little horse crossing the river. The second show the next bumping in the river. These scenarios need complex knowledge to explain. This section shows how the story can take place. To represent the process neatly in the paper, the figures are simplified. Attributes are not represented as a node in the figures. Unrelated nodes are ignored too.

The beginning of the first scenario can be described as Fig. 6. Nodes such as bump and river are regarded as initial nodes. Initial relations are also given out. The reasoning process can be shown in the next figure.

In the second scenario, some activated nodes such as happy should be brought in. The process can be represented as figure 8.

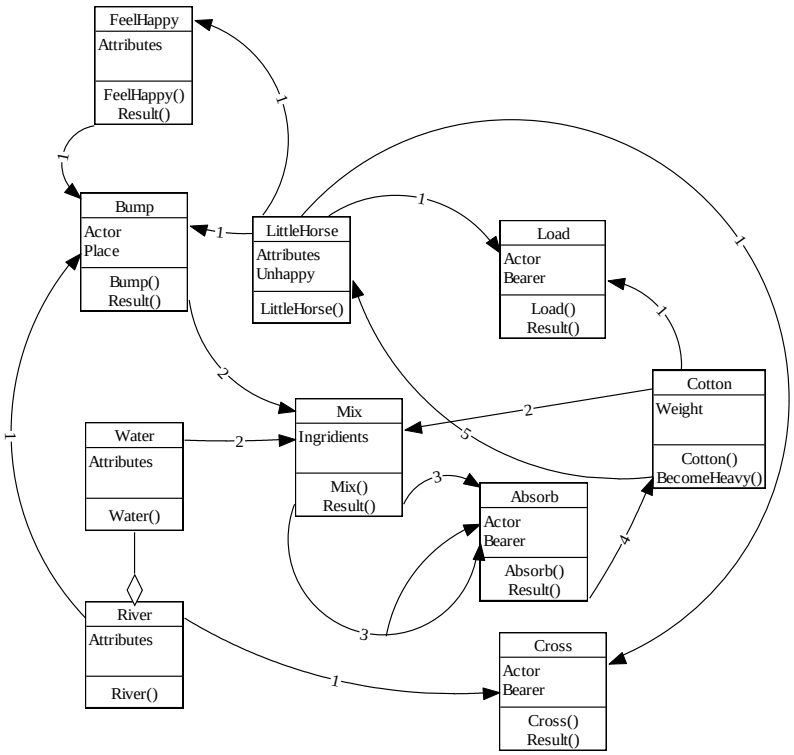


Fig. 8. This figure represents the bump happened the next day. Because the horse fell happy, it repeated the same bump in the same river. What's different is cotton can absorb water and become heavy. That cause the little unhappy.

The story can be explained by the soaking the existing knowledge begins with the given nodes. The soaking results should include the explaining of the target scenario.

5. Problem Solving Algorithm

5.1. Reasoning Process

Before problem solving, known scenarios are analyzed. A directed graph of reasoning is set up according the concept networks represent them. Nodes of concept network are regarded as nodes of the graph. Known relations are added to the graph as edges. Then new nodes are found by searching the knowledge network and activated. When activated, they become known nodes and take part in the next round of soaking. When all the known nodes are activated, a reasoning graph is built which is called G. If the reasoning graph of result scenario is the sub graph of G, G can map into a concept network which represent the answer of the problem.

5.2. Naïve Algorithm

The simplest method is to soak in a hierarchical way until all needed nodes in the result graph are activated. The algorithm may cost a lot of time and space, but the output can include the full process of the scenario. That's to say the scenario can be explained by the algorithm. Let CNBegin and CNresult be the scenarios for the concept networks represent the beginning status and the expected result. Let Ni be nodes of the joint of SBegin and Sresult. Let Ei be the edges of CNBegin. Let G be the result graph. Let Gresult be the reasoning graph of G. Fig. 9 shows the algorithm.

After application of Algorithm 1, a graph G is returned as the answer. The answer is a solution of the problem, but not a simple one.

5.3. Deduction

To find a simple answer of the problem, redundant nodes and edges should be removed from G. For the deduction process, key nodes are defined as the initial nodes of the problem graph. Nodes other than key nodes are called normal nodes. They are activated in the soaking process. If a node can't activate any key nodes, it's unnecessary for the problem solving. Figure 10 shows the deduction algorithm.

With the algorithm described in Fig. 10, unnecessary nodes can be deleted to make the representation simple and clear.

```
Algorithm 1:
ProblemSolve(CNBegin,CNResult)
{
  //Initialization
  for all Ni in CNBegin or CNResult
     if(Ni∉G.Nodes) G.AddNode(Ni);
   for all Ei in SBegin
     if(Ei∉G.Edges) G.AddEdge(Ei);
  //Soak
  while(GResult is not a subgraph of G)
     {
      for all Nj in G
        {
         for all Nk as parent concept of Nj
            {
             Nk.Activated=True;
             G.AddNode(Nk);
             G.AddEdge(Nk,Nj);
            }
          for all Nk as part of Nj
            {
             Nk.Activated=True;
             G.AddNode(Nk);
             G.AddEdge(Nk,Nj);
            }
           //For activate operation with more than one parameters, Try all combinations
           for all Ns in G
             if(Ns can be parameter of Nj)Nj.Activate(Ns);
         }
        }
  return(G);
}
```

Fig. 9. The Soaking Algorithm

6. Experiments and Results

Because the evaluation method can't be reasonably defined, heuristic algorithms and genetic algorithms can't be applied effectively. Only traditional state-space searching algorithm can be compared with soaking method. Hops and soaked nodes can reflect the complexity of soaking algorithm while potential states can be regarded as the complexity of traditional state-space searching algorithm.

We tried some dense monotonic scenarios with the soaking method and state space searching method. Any value change of any object can cause a new state. Because continuous value can cause infinite states, they were ignored in the experiments. The result is presented as below:

Algorithm 2:

```
Deduct(G)
{
Node Neighbor;
int KeyPaths;
 For all normal node Ni in G
    {KeyPaths=0
     for all neighbor nodes Nj of Ni
        {Neighbor=Nj;
          if(FindKey(Nj))KeyPaths=KeyPaths+1;
         }
      if(KeyPaths<1)G.DeleteNode(Ni);
     }
}

Boolean FindKey(Node)
{
  if(Node is a key node)
     return ture;
  else
     FindKey(Node.NotVisitedNeighbor);
}
```

Fig. 10. The Deduction Algorithm

Table 1. Result of Soaking Method and Traditional State-space Searching Method in Scenario Comprehension Tasks.

Problem	Hops/Soaked Nodes of Soaking Method	Number of Potential States for State space Searching Method
Salt solute in water.	2Hops/36Nodes	64
A bag of salt drop into water.	4Hops/257Nodes	9216
A little horse with a bag of salt pass a little river and felt into it, then the horse felt happy.	8Hops/1026Nodes	Not available

From the result of three simple problems, we can see state space searching method can cause sharp increase of complexity in dense knowledge systems. What's more, if all elements of the problem are not explicitly stated, the complexity may be related to all the details of knowledge base. It means the problem can't be solved by with such methods. Object-

oriented soaking method can solve such problems in dense knowledge environment.

7. Discussion and Future Work

7.1. On Knowledge Network and Complexity

Object oriented method provide rich semantics. For state searching algorithms, every value of any concept may course a change of state. Because the total number of object is unknown, states of the problem are unknown. Let totally N concepts and objects concerned in the problem and every concepts and objects have n values. The number of states is n^N. Be aware that N should be an unknown great number. Even heuristic algorithms can't handle this in limited time.

The basic element of this method is object, so the number nodes become N*n. Because only a few objects and concepts can be activated, N can be a relatively small number. Provide the answer is fixed, when the knowledge network become more complex, the method become slower. But more simple answer may be found.

7.2. Enhance the Method with Association Rules

As experience can help people find answer quickly, mining results such as association rules can reduce object activation or even find answer directly. This is a promising way to enhance the method.

7.3. More Flexible Methods

In this paper, only definite activation is used to activate nodes. Other mechanisms such as association need to activate nodes with similar features. These mechanisms may be concerned in future works.

8. Conclusion

Inspired by human's association, a naive object-oriented method is advanced to comprehending how things happened without knowing all the states. By soaking knowledge network, the process is discovered and represented as a reasoning graph. The method is much less complex than state space searching methods in dense knowledge systems.

Acknowledgements. This work was supported by 973 Program(Project No. 2010CB327900) and Shanghai Science and Technology Development Funds(Project No. 08511501703).

References

1. Alexander Reinefeld, and Peter Ridinger: Time-Efficient State Space Search. Artificial Intelligence Vol. 71, No. 2, 397-408. (1994)
2. Meuleau Nicolas et al.: A heuristic search approach to planning with continuous resources in stochastic domains. Journal of Artificial Intelligence Research, Vol. 34, 27-59. (2009)
3. Rüdiger Ebendt and Rolf Drechslerb: Weighted A* search – unifying view and application. Artificial Intelligence, Vol. 173, No. 14, 1310-1342. (2009)
4. Yngvi Bj¨ornsson, Vadim Bulitko and Nathan Sturtevant: TBA*: Time-Bounded A*. In proceeding of IJCAI2009, 431-436 (2009)
5. K. Araki, S. Gnesi, and D. Mandrioli: Using Abstractions for Heuristic State Space Exploration of Reactive Object-Oriented Systems. In Proceeding FME 2003, 462-481 (2003)
6. Jean Piaget: Structuralism. New York: Basic Books, USA. (1970)
7. Adrian Cussins: The Connectionist Construction of Concepts. The Philosophy of Artificial Intelligence. Oxford University Press, UK. (1990)
8. Jonathan Lawry: A framework for linguistic modeling. Artificial Intelligence, Vol. 155, 1–39. (2004)
9. LinXiang Jia: Connectionist Cognitive Psychology. Shanghai Education Press, China. (2006)
10. Fodor J. A. and Pylyshyn Z. W.: Connectionism and Cognitive Architecture: A Critical Analysis. Cognition, Vol. 28, 3-71. (1988)
11. David Barner, Shunji Inagaki and Peggy Li: Language, thought, and real nouns. Cognition, Vol. 111, 329–344. (2009)
12. M. A. K. Halliday: An Introduction to Functional Grammar. HODDER ARNOLD. (2008)
13. Michael Eysenck and Mark T. Keane: Cognitive Psychology: A Student's Handbook. Psychology Press, a member of the Taylor & Francis Group. (2000)
14. Lokendra Shastri: Default reasoning in semantic networks: A formalization of recognition and inheritance. Artificial Intelligence, Vol. 39, No. 3, 283-355. (1989)
15. Hai Zhuge: Communities and Emerging Semantics in Semantic Link Network: Discovery and Learning. IEEE Transactions on Knowledge and Data Engineering, Vol. 21, No. 6, 785-799. (2009)
16. Michael Freund: On the notion of concept I. Artificial Intelligence, Vol. 172, 570–590. (2008)
17. John F. Sowa: Knowledge Representation: Logical, Philosophical, and Computational Foundation. Machine Press, China. (2003)
18. Chenghua Dang et al.: Wordnet-based summarization of unstructured document. WSEAS Transactions on Computers, Vol. 7, No. 9, 1467-1472. (2008)
19. Douglas B. Lenat: CYC: A Large-Scale Investment in Knowledge Infrastructure. COMMUNICATIONS OF THE ACM, Vol. 38, No. 11, 33-38. (1995)
20. Xixu Fu and Hui Wei: Cognition Inspired Object Oriented Knowledge Warehouse Architecture. Journal of Software, Vol. 5, No. 9, 926-933. (2010)
21. Adelman, JS and Brown, GDA.: Modeling lexical decision: The form of frequency and diversity effects. Psychological Review, Vol. 115, 214-227. (2008)

22. Zdravkovic M, Trajanovic M.: Integrated Product Ontologies for Inter-Organizational Networks. COMPUTER SCIENCE AND INFORMATION SYSTEMS, Vol. 6, No. 2, 29-46. (2009)
23. Garcez Artur S. D'Avila, Gabbay, Dov M. and Lamb Luis C.: Value-based argumentation frameworks as neural-symbolic learning systems. Journal of Logic and Computation, Vol. 15, No. 6, 1041-1058. (2005)
24. Cruse D. A.: Language, Meaning and Sense: Semantics, An Encyclopedia of Language. Routledge, London. (1990)
25. Barber C.: The English Language: A Historical Introduction. Cambridge University Press, UK. (1993)
26. Otto Jespersen: The philosophy of grammar. Beijing World publishing company, China. (2008)
27. P.J Hayses: The naive physics manifesto. The Philosophy of Artificial Intelligence, Oxford University Press, UK. (1990)
28. Anthony Morse and Malin Aktiusa: Dynamic liquid association: Complex learning without implausible guidance. Neural Networks, Vol. 22, No. 7, 875-889. (2009)
29. Angelo Cangelosi: Evolution of Communication and Language Using Signals, Symbols, and Words. IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, Vol. 5, No. 2, 93-101. (2001)
30. Forsythe D.E., Buchanan B.G.: Knowledge acquisition for expert systems: some pitfalls and suggestions. IEEE Transaction on Systems, Man and Cybernetics, Vol. 19, No. 3, 432-435. (1989)

Xixu Fu was born in 1981. He received his master degree in 2007. Now he is a Ph.D Candidate in Fudan university. His research interests include artificial intelligence, cognitive science and data mining.

Hui Wei was born in 1971. He received his PhD in department of computer science at Beijing University of Aeronautics and Astronautics in July 1998. From September 1998 to November 2000 he was a postdoctoral in the department of computer science and institute of artificial intelligence at Zhejiang university. In November 2000 he jointed the department of computer science and engineering at Fudan University. He is currently a professor in Fudan university. His research interests include artificial intelligence and cognitive science.

Received: September 15, 2010; Accepted: February 17, 2011.

DOI: 10.2298/CSIS100322028W

Research on Discovering Deep Web Entries

Ying Wang[1,2], Huilai Li[3], Wanli Zuo[1,2], Fengling He[1,2], Xin Wang[1,4], and Kerui Chen[1,2]

[1] College of Computer Science and Technology, Jilin University, 130012 Changchun, China
[2] Key Laboratory of Computation and Knowledge Engineering, Ministry of Education, China
{wangying2010, zuowl, hefl}@jlu.edu.cn, Chenke0616@163.com
[3] College of Mathematics, Jilin University, 130012 Changchun, China
lihuilai@jlu.edu.cn
[4] College of Software, Changchun Institute of Technology, 130012 Changchun, China
wangxccs@126.com
Wanli Zuo, zuowl@jlu.edu.cn

Abstract. Ontology plays an important role in locating Domain-Specific Deep Web contents, therefore, this paper presents a novel framework WFF for efficiently locating Domain-Specific Deep Web databases based on focused crawling and ontology by constructing Web Page Classifier(WPC), Form Structure Classifier(FSC) and Form Content Classifier(FCC) in a hierarchical fashion. Firstly, WPC discovers potentially interesting pages based on ontology-assisted focused crawler. Then, FSC analyzes the interesting pages and determines whether these pages subsume searchable forms based on structural characteristics. Lastly, FCC identifies searchable forms that belong to a given domain in the semantic level, and stores these URLs of Domain-Specific searchable forms to a database. Through a detailed experimental evaluation, WFF framework not only simplifies discovering process, but also effectively determines Domain-Specific databases.

Keywords: Deep Web, ontology, WPC, FSC, FCC.

1. Introduction

With the rapid development of the web, more and more information has been transferred from static web pages (that is Surface Web) into web databases (that is Deep Web) managed by web servers[1][2]. As Fig.1 conceptually illustrates, on this so-called “Deep Web”, numerous online databases provide dynamic query-based data access through their query interfaces, instead of static URL links[3]. The data in Deep Web are of great value, but difficult to

query and search. With new web databases added and old web databases modified and removed constantly, artificial classification is a laborious and time-consuming task, so it is imperative to accelerate research on discovering effectively which searchable databases are most likely to contain the relevant information for which a user is looking.

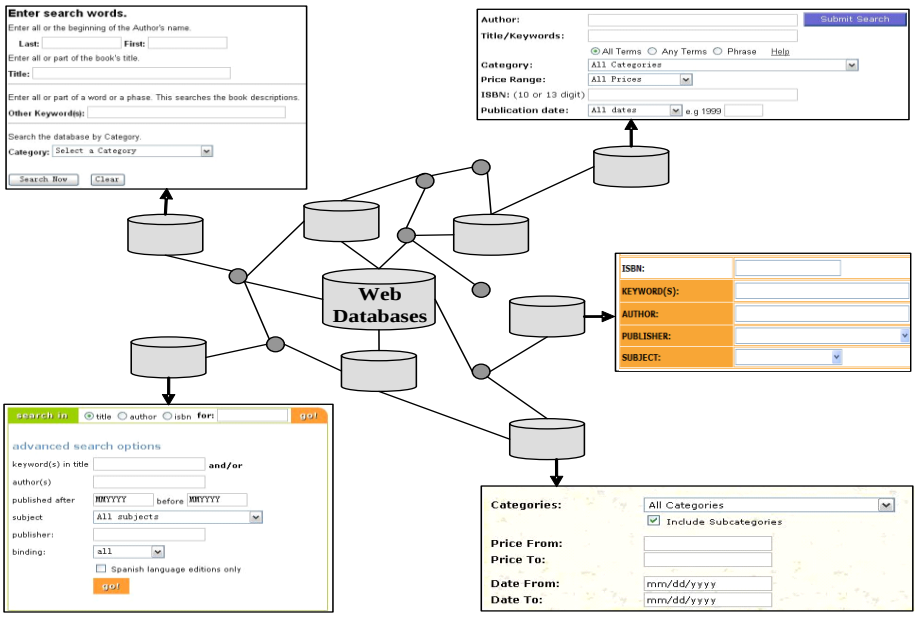


Fig. 1. Deep Web provides dynamic query-based data access through their query interfaces

Discovering Deep Web entries is the first significant step in integrating Deep Web data, in order to assist users accessing Deep Web, recent efforts have focused on two kinds of approaches to discover Deep Web entries automatically: Pre-Query and Post-Query[4].

Pre-Query identifies web databases by analyzing the wide variation in content and structure of forms. In 2005, Barbosa L and Freire J.[5] propose a crawling framework FFC to automatically locate Deep Web databases by focusing the search on a given topic; by learning to identify promising links; and by using appropriate stop criteria that avoid unproductive searches within individual sites. However, this method has some limitations: it requires substantial manual tuning and the form set retrieved by FFC is very heterogeneous. After two years, Barbosa L and Freire J.[6][7][8] present again a new framework ACHE that addresses these limitations, which automatically and accurately classifies online databases based on features that can be easily extracted from web forms. Manuel Alvarez et al.[9] provide the architecture of DeepBot, a prototype of hidden-web focused crawler able to access Deep Web content. Their approach is based on a set of domain

definitions, each one describing a data-collecting task. From the domain definition, the system uses several heuristics to automatically identifying relevant query forms. Hui Wang and Wanli Zuo[10] propose a three-step framework to automatically identify domain-specific hidden Web entries. With those obtained query interfaces, they can be integrated to obtain a unified interface which is given to query for users. Li Yingjun et al.[11] propose a Domain-Oriented Deep Web data source Discovery method (DO-DWD) and a novel Domain Identification strategy of Deep Web data sources (DIDW). In the discovery stage, using machine learning algorithms and some heuristic rules to find query interfaces of the data sources; In the identification stage, identifying Deep Web data sources associated with the domain by calculating the relevance between a query interface and the domain based on semantic similarity. Pengyi Zhang et al.[12] propose a novel hybrid approach to construct a collection of government Deep Web resources. It combines automatic computation power and human intelligence through social computing. This approach presents the opportunity of building information structures on deep web portals in a scalable and sustainable manner. However, most of the above approaches do not consider applying background knowledge, which is important to understand problems and situations.

Post-Query approach identifies web databases from the retrieved results by submitting probing queries to the forms. In 2003, Luis Gravano and Panagiotis G.Ipeirotis[13] introduce QProber, a modular system that automates the classification process by using a small number of query probes, generated by document classifiers. However, this approach relies on a pre-learned set of queries for database classification. Additionally, if new categories are added or old categories removed from the hierarchy, new probes must be learned and each source re-probed. After five years, Luis Gravano and Panagiotis G.Ipeirotis[14] present a novel "focused-probing" sampling algorithm that detects the topics covered in a database and adaptively extracts documents that are representative of the topic coverage of the database. However, if the topic is not self-contained, then it will affect the database selection. Victor Z.Liu, et al.[15] develop a probabilistic approach to use dynamic probing(issuing the user query to the databases on the fly) in a systematic way, so that the correctness of database selection is significantly improved while the meta-searcher contacts the minimum number of databases. However, when the user does not care about the answer's correctness, the method will not applicable. Lu Jiang et al.[16] propose a novel Deep Web crawling method with Diverse Features. They thought that the key to Deep Web crawling was to submit promising keywords to query form and retrieve Deep Web content efficiently. Keywords are encoded as a tuple by its linguistic, statistic and HTML features so that a harvest rate evaluation model can be learned from the issued keywords for the un-issued in future. One year later, Lu Jiang et al.[17] propose a novel Deep Web crawling framework based on reinforcement learning, in which the crawler is regarded as an agent and deep web database as the environment. The agent perceives its current state and selects an action (query) to submit to the

environment according to Qvalue. The framework not only enables crawlers to learn a promising crawling strategy from its own experience, but also allows for utilizing diverse features of query keywords. However, it is some of wasting network and server resources by submitting a large number of queries only for the purpose of classification.

From the analysis above, Post-Query approach cannot be adapted to structured multi-attribute forms[18], so it is difficult for Post-Query approach to obtain better classification effects. Therefore, the method of Pre-Query which depends on visual features of searchable forms, namely, attribute labels and other available resources, are able to deal with highly heterogeneous form sets and usually used to indicative the database domain. That is to say, the discovery of Deep Web entries can be translated into the issue of distinguishing query forms. In this paper, we apply the Pre-Query approach for automatically classifying Domain-Specific forms by importing focused crawling and ontology technique. The paper is organized as follows: The section 2 presents the overview of discovering Deep Web entries, which includes problem formulation and WFF framework. The section 3 presents the process of WFF framework during discovering Deep Web entries. The section 4 presents the experiment results of WFF framework. Finally, in section 5, conclusions are drawn and future work is considered.

2. Overview

2.1. Problem Formulation

***Definition1*.** Deep Web Database: a Deep Web database is a web site, which contains searchable forms and a back-end database. Each database has specific searchable forms and result pages, generally, each searchable form is also known as "Input Schema", and result pages are known as "Output Schema", therefore, a database can be described as a triple-tuple (ds , IS , OS):

(1) ds denotes the back-end database behind a web site, which runs on web server.

(2) IS denotes a searchable form schema of web database, $IS = \{a_1, a_2, ...a_n\}$, where $a_i (0 \le i \le n)$ denotes a semantic attribute.

(3) OS denotes the result pages which are obtained by submitting requests from searchable forms.

***Definition2*.** Domain-Specific Database Discovery: It is used to judge whether a target database is relevant to the source database. Given a Deep Web source set $DS = \{ds_1, ds_i, ...ds_n\}$ and a category set

$C = \{C_1, C_2, ..., C_m\}$. Domain-Specific database discovery can be regarded as a mapping function from relational databases to the "best" category, namely formula (1):

$$f : DS \to C \tag{1}$$

The mapping function can make each database ds_i $(1 \le i \le n)$ from DS assign to a specific category C_j $(1 \le j \le m)$.

The fact that Deep Web sources are sparsely distributed makes especially challenging on locating them according to different domains[19]. There are mainly four questions:

***Qustion1*.** How to find "entries" to Deep Web databases? The entry of each Deep Web database is the query interface(searchable form). To access a web database, we must firstly find its searchable form.

***Qustion2*.** Which depth does each searchable form locate in a site? The depth of each searchable form is the minimum number of hops from the root page to the page which contains the searchable form.

***Qustion3*.** How to recognize the searchable forms of Deep Web databases? Accessing to databases is provided only through restricted forms, not all the HTML forms are interfaces of Deep Web sites. HTML forms can be divided into searchable forms and non-searchable forms, searchable forms are query interfaces.

***Qustion4*.** How to distribute the subject of web databases? There are great subject diversities among web databases, it is important to locate Domain-Specific databases.

Therefore, discovering topic relevant Deep Web entries accurately is one of the critical steps toward the integration of heterogeneous Deep Web sources.

2.2. WFF Framework

Since ontology is a well-formed knowledge representation, to access Deep Web effectively, we present a novel framework WFF for effectively locating Deep Web entry points based on focused crawling and ontology technique. WFF framework given in Fig. 2 consists of three main components: Web Page Classifier(WPC), Form Structure Classifier(FSC) and Form Content Classifier(FCC).

Firstly, WPC discovers potentially interesting pages based on ontology-assisted focused crawler. Then, FSC analyzes these interesting pages and determines whether these pages subsume searchable forms based on structural characteristics. Lastly, FCC identifies searchable forms that belong to a given domain in the semantic level, and stores these URLs of Domain-Specific searchable forms to a database. Discovering Deep Web entries is simplified by combining three hierarchical classifiers, which makes the overall classification process more accurate and robust.

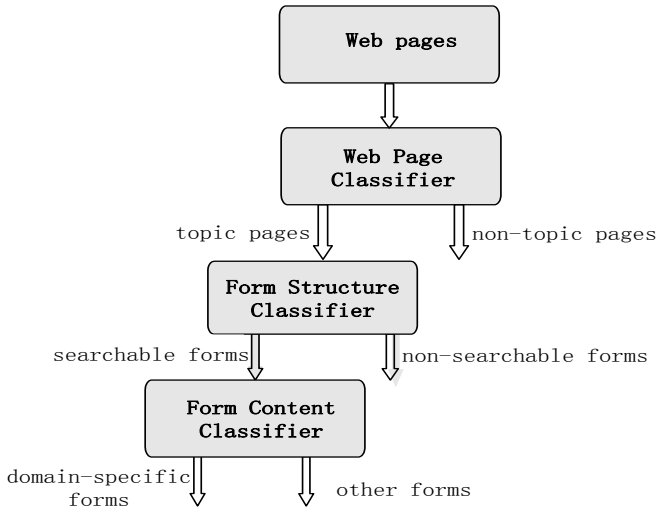


Fig. 2. WFF framework for discovering Deep Web entries, which contains Web Page Classifier, Form Structure Classifier and Form Content Classifier.

3. WFF Framework for Discovering Deep Web Entries

3.1. Ontology

Ontology as the foundation of knowledge processing, a concept model describing information system in semantic and knowledge level, user's queries and relevant data can be mapped to ontology, in this way, ontology can be seen as a knowledge system which describes concepts and relationships[20].

Definition3. Domain Ontology Concept Model(DOCM): DOCM is a data model that describes a set of concepts and relationships that may appear in a specific domain. It should be understandable by machine so that it can be used to reason about these objects within that domain. Each object can be denoted as $Class = \{CM, DT, \{S_i\}, \{CA_i\}, \{SC_i\}\}$, which describes the relevant information of object.

CM : The main class of object, which is universal and easy to understand for users. It can be seen as the keyword of object.

DT : The data type of object, such as "string", "numerical" and so on.

$\{S_i\}$: The synonymous set of CM , namely, the concept aliases.

$\{CA_i\}$: The condition property set of object, which is "Part-Of" relationship to CM .

$\{SC_i\}$: The sub class set of CM , which is "Is-A" relationship to CM .

DOCM has a good organizational structure, which represents high-level background knowledge with concepts and relationships[21]. In this paper, the concepts and relationships of DOCM are extracted from searchable forms and result pages, and the ontology is implemented by Protégé API and represented in the Web Ontology Language(OWL)[22]. To operate ontology is equivalent to operate the OWL file.

An example of Book-Domain ontology is shown in Fig. 3.

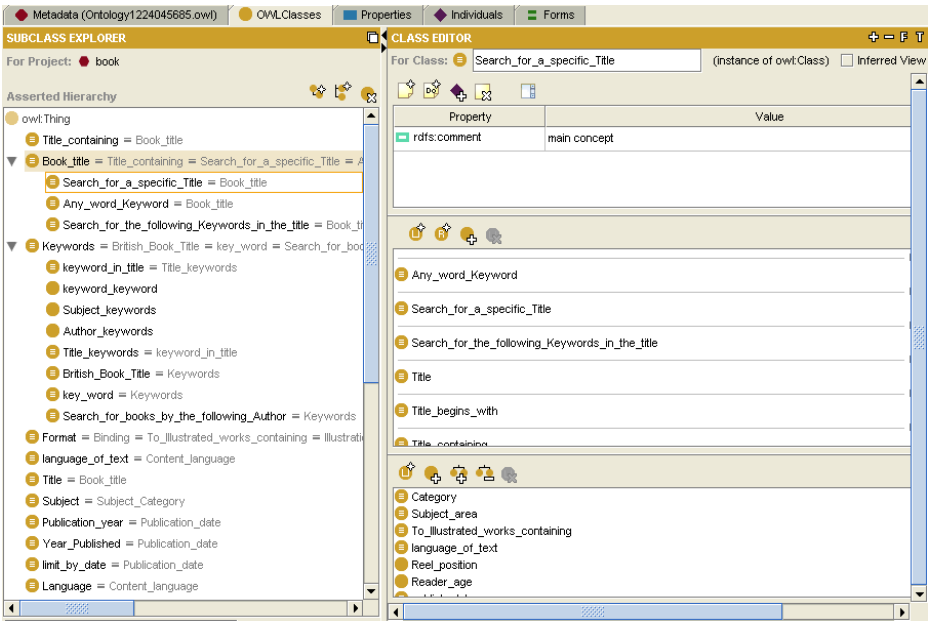


Fig. 3. An example of Book-Domain ontology, which describes the concepts and the logical relationships using a hierarchical tree structure.

3.2. WPC

WPC, namely, ontology-based focused crawling, which is used to guide the crawler and focus the search on interesting pages by analyzing features of web pages[23]. K. C.-C. Chang et al.[24] point out that the depth of Deep Web searchable form is less than 5, 94% of the searchable form depth is less

than 3. Therefore, when locating an interesting page, the crawler will comply with two strategies:

Strategy1 The ontology-based crawler follows the hyperlinks from the page which is classified as being on topic.

Strategy2 The ontology-based crawler follows hyperlinks only to specific levels of depth.

Definition4. Page Similarity: Suppose $\vec{D}$ is a page feature vector containing m feature terms, $\vec{D} = \{(k_{1,d}, w_{1,d}), (k_{2,d}, w_{2,d}), ...(k_{m,d}, w_{m,d})\}$, $\vec{q}$ is a topic vector containing n feature terms, $\vec{q} = \{(t_{1,q}, w_{1,q}), (t_{2,q}, w_{2,q}), ...(t_{n,q}, w_{n,q})\}$. If these terms in page feature vector and topic vector can be found in ontology, then finding these corresponding concepts of terms from ontology, and replacing these terms with their corresponding concepts. These terms in page feature vector and topic vector can not be found from ontology called unlogin terms. After replacing these terms, page feature vector $\vec{D}$ can be divided into page concept vector $\vec{PCV}$ and page unlogin term vector $\vec{PUV}$, topic vector $\vec{q}$ can be divided into topic concept vector $\vec{TCV}$ and topic unlogin term vector $\vec{TUV}$.

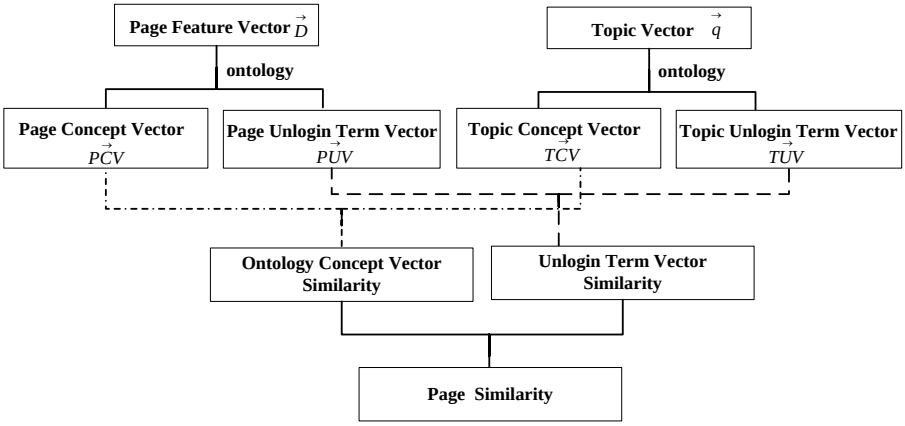


Fig. 4. The structure of page similarity computation, which contains ontology concept vector similarity and unlogin term vector similarity.

If several terms are matched with the same ontology concept, then replacing these terms with this concept, and summing these weights of several terms as the corresponding concept weight. The similarity between page feature vector $\vec{D}$ and topic vector $\vec{q}$ can be calculated in formula(2):

$$Sim(\vec{D}, \vec{q}) = \alpha \cdot Sim_{st\_ontology}(\vec{PCV}, \vec{TCV}) + (1-\alpha) \cdot Sim_{unlogin}(\vec{PUV}, \vec{TUV}) \quad (2)$$

Where α is an impact factor, whose role is to adjust the impact to similarity between page concept vector $\vec{PCV}$ and page unlogin term vector $\vec{PUV}$. The structure of page similarity computation is shown in Fig. 4.

If a page which contains hyperlinks is topic relevant by page similarity algorithm, then we need to extract hyperlinks from the page and analyze the topic relevance of these hyperlinks, else, abandoning these hyperlinks.

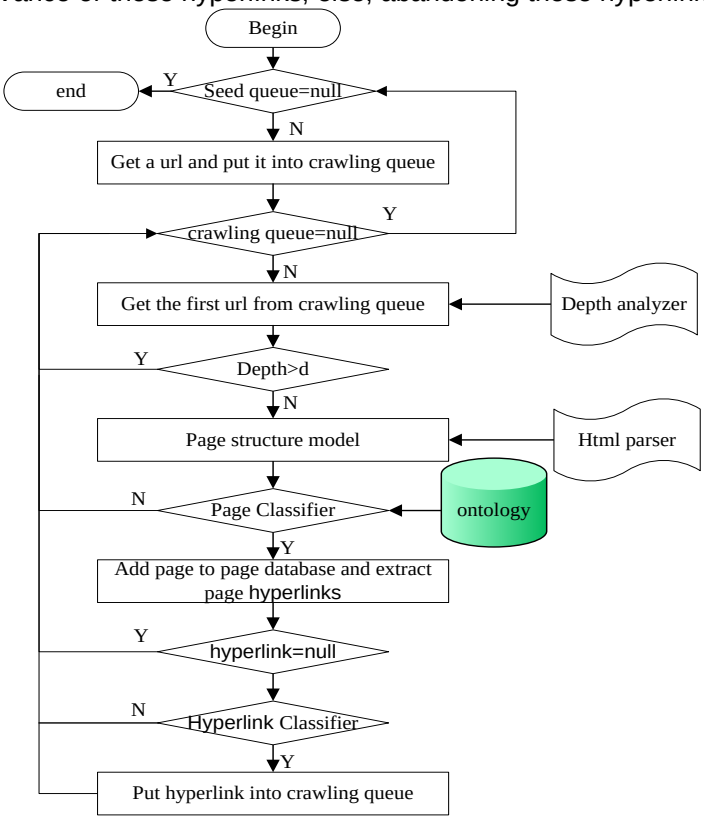


Fig. 5. WPC executive process: WPC receives as input a set of "seed" pages and recursively obtain new ones by following hyper-links in the standard depth-first traversal, lastly, recording interesting pages into repository and calling FSC.

Definition5. Hyperlink similarity: Extracting the anchor from topic page $\vec{D}$ to generate hyperlink anchor vector $\vec{Anchor} = \{(l_{1,link}, w_{1,link}), (l_2, w_{2,link})...(l_{k,link}, w_{k,link})\}$, and then calculating the anchor similarity $Sim(\vec{Anchor}, \vec{q})$ between anchor vector $\vec{Anchor}$ and topic vector $\vec{q}$ by page similarity method. The final hyperlink similarity can be calculated in formula(3):

$$Sim_{link}(\vec{Anchor}, \vec{q}) = \beta Sim(\vec{D}, \vec{q}) + (1-\beta) Sim(\vec{Anchor}, \vec{q}) \quad (3)$$

Where β is an impact factor, whose role is to adjust the impact to similarity between page feature vector $\vec{D}$ and anchor vector $\vec{Anchor}$.

The process of WPC is shown in Fig. 5.

3.3. FSC

Definition6. Searchable form: The form characterized by its capacity of submitting a query to an online database. When a user submits queries in the searchable form, the queries will be issued against the database and return the results of query execution.

Definition7. Non-searchable form: The form which does not represent database queries, for example, login forms, registration, mailing list subscriptions forms, email forms and so on.

FSC uses decision tree classifier which is proved to have lowest error rate[25]. Decision Tree algorithm is used to build the classifier of form structure for filtering out non-searchable forms and ensures only searchable forms that can be added to the form database.

Definition8. Decision Tree: A Decision Tree is a decision support tool which uses a tree-like graph or model of decisions and their possible consequences. Each internal node tests an attribute, each branch corresponds to attribute value, and each leaf node assigns a classification[26][27].

C4.5 is an algorithm used to generate a Decision Tree developed by Ross Quinlan[28]. At each node of the tree, C4.5 chooses one attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. Its criterion is the normalized information gain that results from choosing an attribute for splitting the data. The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recurses on the smaller sublists[29]. The information gain of attribute A_i is calculated with formula(4):

$$Gain(D, A_i) = Entropy(D) - Entropy_{A_i}(D) \quad (4)$$

Where D is the training examples, A_i is the splitting attribute. The information gain is based on entropy function from information theory, which is denoted in formula (5):

$$Entropy(D) = -\sum_{j=1}^{|C|} \Pr(c_j) \log_2 \Pr(c_j) \quad (5)$$

Where $\Pr(c_j)$ is the probability of class c_j in training examples D, which is the number of examples of class c_j in D divided by the total number of

examples in D, $\sum_{j=1}^{|C|}\Pr(c_j)=1$. If the number of possible values of the attribute A_i is v, and using A_i to partition the data D, we will divide D into v disjoint subsets $D_1, D_2 ... D_v$. The entropy after the partition by attribute A_i is shown in formula (6)[30]:

$$Entropy_{A_i}(D)=\sum_{j=1}^{|v|}\frac{|D_j|}{|D|}\times Entropy(D_j) \quad (6)$$

C4.5 Decision Tree algorithm is as follows:

C4.5 Decision Tree algorithm

```
Input: Training_examples D, attribute_list
Output: decision_tree
BEGIN
Generate_decision_tree(D, attribute_list)
1. Initialize()
2. creatNode(N)
3. if(Training_examples=null)
4.    return N="failure"
5. if(Training_examples∈C)
6.    return leafNode(N)=C
7. if(attribute_list=null)
8.    return leafNode(N)=M(C)
9. for(each A_i ∈ attribute_list)
10.   if( A_i is continuous)
11.      splitting( A_i )
12.      GrainRatio=compute( A_i )
13. selectMaxGrainRatio( A_i )
14. leafNode(N)= A_i
15. for each value d of A_i
16.   addCondition( A_i = d )
17.   if( D_i = φ ) // D_i is the subset of D based on the d value of A_i
18.      addLeafNode( N' )=M(C)
19.   else
20. return Generate_decision_tree( D_i ,attribute_list)
END
```

The generated Decision Tree is shown in Fig. 6. Decision Tree builds an interpretable model that represents a set of rules.

```
decision_tree:
depth = 1
| isExist Form = No : Non-Searchable form
depth = 1
| isExist Form = Yes
| | depth = 2
| | | attribute-type isExist in AttributeTypeSet = No : Non-Searchable form
| | depth = 2
| | | attribute-type isExist in AttributeTypeSet = Yes
| | | | depth = 3
| | | | | special-attribute-numbers >= 3 = No : Non-Searchable form
| | | | depth = 3
| | | | | special-attribute-numbers >= 3 = Yes
| | | | | | depth = 4
| | | | | | | isExist SubmitButton = No : Non-Searchable form
| | | | | | depth = 4
| | | | | | | isExist SubmitButton = Yes
| | | | | | | | depth = 5
| | | | | | | | | is ButtonTypeSubmit = No : Non-Searchable form
| | | | | | | | depth = 5
| | | | | | | | | is ButtonTypeSubmit = Yes
| | | | | | | | | | depth = 6
| | | | | | | | | | | isExist QueryKeywordSet in {name,value} = Yes : Searchable form
| | | | | | | | | | | isExist QueryKeywordSet in {name,value} = N0 : Non-Searchable form
| | | | | | | | | is ImageTypeSubmit = Yes
| | | | | | | | | | depth = 6
| | | | | | | | | | | isExist QueryKeywordSet in {name,value,alt,src} = Yes : Searchable form
| | | | | | | | | | | isExist QueryKeywordSet in {name,value,alt,src} = N0 : Non-Searchable form
```

Fig. 6. From the Decision Tree, we can obtain the rules for classifying searchable forms and non-searchable forms.

The rules extracted from Decision Tree are as follows:

Rule1: If there is no <Form> tag in a page, then this page is non-searchable form.

Rule2: If there exists <Form> tag, then extracting attribute types between <Form> and </Form>. If each attribute type does not exist in “Attribute Type Set”, then this page is non-searchable form.

Rule3: If there exists <Form> tag, and there are attribute types in “Attribute Type Set”. If “Attribute Number” is less than 3, then this page is non-searchable form.

Rule4: If there exists <Form> tag, and there are attribute types in “Attribute Type Set”, “Attribute Number” is more than 3, but there is no submit button, then this page is non-searchable form.

Rule5: If there exists <Form> tag, there are attribute types in “Attribute Type Set”, “Attribute Number” is more than 3, and there exists submit button with “submit” type, but “Button Marker” does not exist in “Search Word Set”, then this page is non-searchable form.

Rule6: If there exists <Form> tag, there are attribute types in “Attribute Type Set”, “Attribute Number” is more than 3, and there exists submit button with “image” type, but “Image Marker” does not exist in “Search Word Set”, then this page is non-searchable form.

Rule7: If there exists <Form> tag, there are attribute types in “Attribute Type Set”, “Attribute Number” is more than 3, there exists submit button with “submit” type, and “Button Marker” is in “Search Word Set”, then this page is searchable form.

Rule8: If there exists <Form> tag, there are attribute types in "Attribute Type Set", "Attribute Number" is more than 3, there exists submit button with "image" type, and "Image Marker" is in "Search Word Set", then this page is searchable form.

FSC based on Decision Tree classifies the searchable forms and non-searchable forms by the above rules.

3.4. FCC

Though FSC, we can find that the topic relevant page contains a searchable form, however, the form content retrieved may belong to a different domain. Therefore, a novel method of ontology-assisted FCC is proposed to identify Domain-Specific databases by analyzing Domain-Specific form content[31][32][33].

Definition9. Ontology assisted FCC: Suppose $\vec{F} = \{(f_{1,d}, w_{1,f}), (f_{2,d}, w_{2,f}), ...(f_{m,d}, w_{m,f})\}$ is a form feature vector containing m form feature terms, where $(f_{i,f}, w_{i,f})$ $(1 \le i \le m)$ denotes a form feature term and its corresponding weight. $\vec{q}$ is the topic vector containing n feature terms $q = \{(t_{1,q}, w_{1,q}), (t_{2,q}, w_{2,q}), ...(t_{n,q}, w_{n,q}),\}$, where $(t_{j,q}, w_{j,q})$ $(1 \le j \le n)$ denotes a topic term and its corresponding weight. Generally, the vocabularies of searchable form are restricted and not duplicated, therefore, we set the weight $w_{i,d}$ of each feature term $t_{i,f}$ as $1/m$. For each feature term $t_{i,d}$ in form d, there are three cases:

Case1 If $t_{i,d} \in DOCM$, then, setting $Sim_i(t_{i,d}, \vec{q}) = 1$.

Case2 If $t_{i,d} \notin DOCM$ and $t_{i,d} \notin \vec{q}$, then, $Sim_i(t_{i,d}, \vec{q}) = 0$.

Case3 If $t_{j,q} \in \vec{q}$, and $t_{j,q} = t_{i,d}$, then, $Sim_i(t_{i,d}, \vec{q}) = \dfrac{w_{i,d} + w_{j,q}}{2}$.

The final similarity between form feature vector $\vec{F}$ and topic vector $\vec{q}$ can be calculated in formula(7):

$$Sim(\vec{F}, \vec{q}) = \frac{\sum_{i=1}^{m} Sim_i(t_{i,d}, \vec{q})}{m} \tag{7}$$

The process of FCC is shown in Fig. 7:

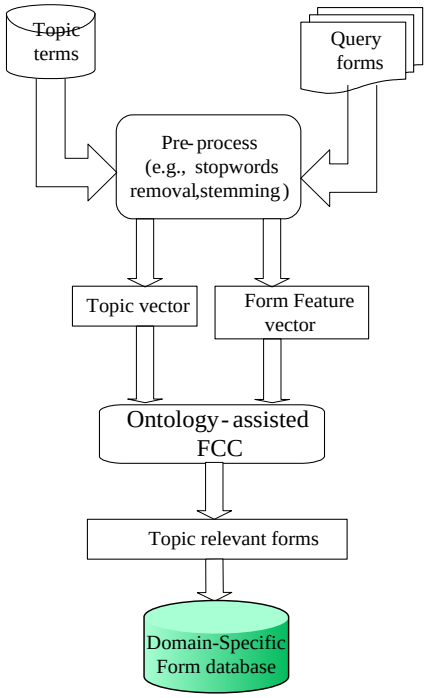


Fig. 7. Ontology plays an important role in recognizing Deep Web entry forms. Therefore, an ontology assisted FCC algorithm was proposed to locate Domain-Specific query interfaces.

4. Experiments

Though the above analysis, we implement the graphical interface for discovering Deep Web entries which is shown in Fig.8.

We evaluate our method with four experiments, respectively, WPC, FSC, FCC and WFF.

Experiment 1 **WPC:** Harvest is usually used to evaluate focused crawling, and it means the fraction of web pages crawled which satisfy the crawling target among the crawled pages. The harvest is shown in formula (8):

$$harvest = \frac{\sum_{p \in P} rel(p)}{|P|} \qquad (8)$$

Where $|P|$ denotes the number of web pages crawled, $rel(p)$ denotes the number of specific topic pages. The initial URLs for the crawler are 100 Book-Domain URLs, which are managed by a manual directory.

Fig. 8 The graphical interface for discovering Deep Web entries

If the impact factor α is set 0.5 in formula(11), namely, they share the same proportion for page concept vector $\vec{PCV}$ and unlogin term vector $\vec{PUV}$, then, though analyzing these 100 Book-Domain pages, the similarity distribution is that 78% pages is more than 0.3, 96% pages is more than 0.25, and 4% pages is less than 0.25, therefore, in most cases, it is more reasonable for setting page similarity threshold(PS) to 0.25 or 0.3. Similarly, the impact factor β is set 0.7 in formula(12), that is to say, we think page similarity is more important than anchor similarity, though analyzing 100 Book-Domain hyperlinks, the similarity distribution shows that 94% hyperlinks is more than 0.25, 97% hyperlinks is more than 0.2, and 3% pages is less than 0.2, therefore, in most cases, it is more reasonable for setting hyperlink similarity threshold(HS) to 0.25. Simultaneously, setting another two parameters: page depth d=4, the maximum number of crawling pages $N = 2000$. We study the performance of WPC by two crawlers with distinct focus strategies: ontology-based focused crawler(OFC) and Best-First focused crawler(BFC)[34]. Best-First focused crawler is based on TF-IDF weight model, though analyzing Book-Domain pages and hyperlinks, in most cases, it is more reasonable for setting page similarity threshold(PS) based on Best-First method to 0.5, and hyperlink similarity threshold(HS) to 0.35. Fig.9 illuminates the performance for OFC and BFC.

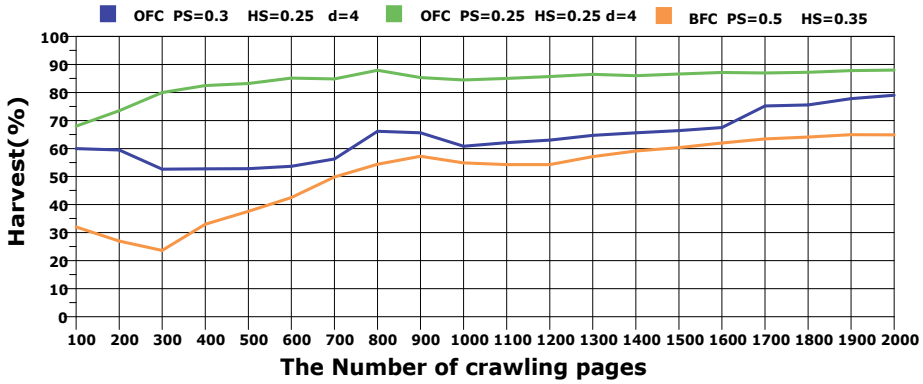


Fig. 9. The result of Web Page Classifier. From the results of WPC, when PS=0.25, it has a higher harvest ratio than PS=0.3. Because that the page similarity for 78% pages is more than 0.3, and 96% pages is more than 0.25, if PS=0.3, it will miss some Domain-Specific pages, so the harvest for PS=0.25 is higher than PS=0.3. whatever page similarity is set 0.25 or 0.3, OFC is performing better with respect to harvest ratio than BFC as the crawling progresses, the substantial increases in harvest ratio is obtained because that OFC relates the crawling topics to the background knowledge base in order to filter out irrelevant web pages.

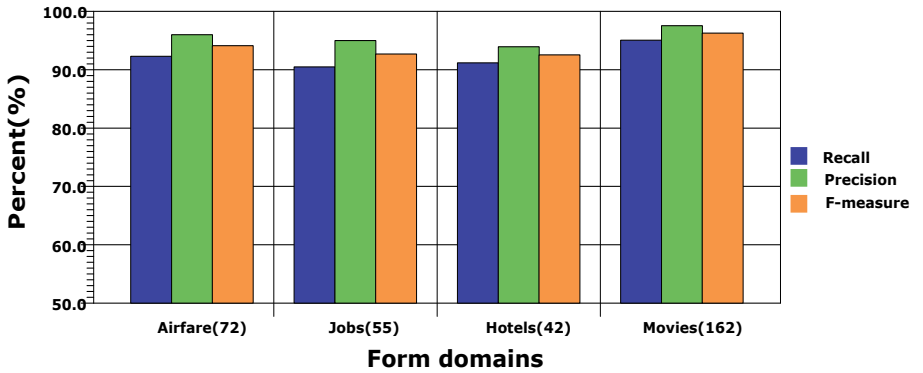


Fig. 10. The results of FSC in different domains, we can see that FSC based on Decision Tree can obtain satisfied accuracy in different domains.

Experiment 2 **FSC:** The evaluation metric for Form Structure Classifier is called Precision, Recall and F-measure. Precision is the percentage of correctly identified searchable forms over all the identified searchable forms by Form Structure Classifier. Recall is the percentage of correctly identified searchable forms over all the searchable forms. F-measure denotes a harmonic mean between precision and recall. In this study, FSC based on Decision Tree is domain-independent, and it is general and can be applied to many different domains. In order to validate FSC, we select four domains from UIUC data set: Airfare, Jobs, Hotels, Movies. The results are shown in Fig.10.

FSC based on Decision Tree can obtain satisfied accuracy. Therefore, the method of FSC based on Decision Tree is feasible.

Experiment 3 **FCC:** The evaluation metric for FCC is also Precision, Recall and F-measure. Precision is the percentage of correctly identified Domain-Specific forms over all the identified Domain-Specific forms by FCC algorithm. Recall is the percentage of correctly identified Domain-Specific forms over all the Domain-Specific forms. F-measure denotes a harmonic mean between Precision and Recall. Similarity threshold setting is a critical step for searchable form classification. There are different results on Recall, Precision and F-measure with different threshold. The threshold is not as small as possible, or the greater the good. In order to better understand the three evaluation metrics, we are on to experiment with different thresholds, which are 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9. The number of selected forms is 160 Book forms. FCC correctness ratio is shown in Fig.11:

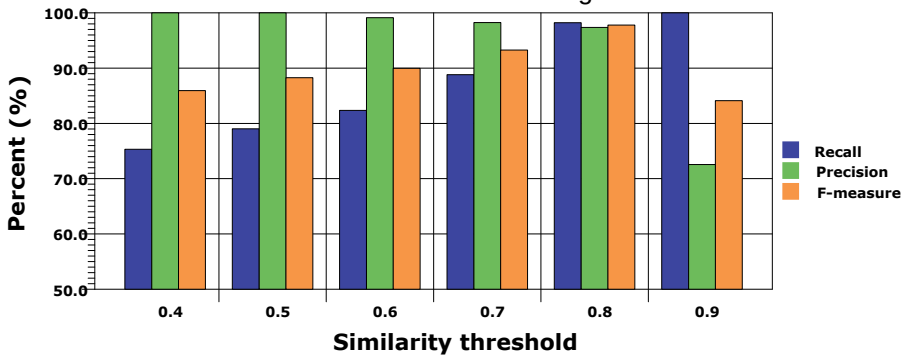


Fig. 11. From the results of FCC, we can see that when the similarity threshold is set low, the results contain most relevant pages, and mistake a lot of irrelevant pages relevant, so Precision is low and Recall is high. When the similarity threshold is set high, it will ignore most relevant pages, so Precision is high and Recall is low. When $\theta = 0.8$, there is a higher accuracy for Recall, Precision and F-measure, therefore, it is more reasonable for $\theta = 0.8$. It also proves that the method of ontology-assisted FCC can identify Domain-Specific forms with high accuracy.

Experiment 4 WFF: If the maximum number of pages for crawler $N = 10000$ and FCC threshold $\theta = 0.8$, then, with the increase of crawling pages, the changes for Domain-Specific forms by OFC and BFC are shown in Fig.12.

Through the detailed analysis above, it indicates that the WFF framework is a scalable alternative to efficiently locate Deep Web entry points based on focused crawling and ontology technique.

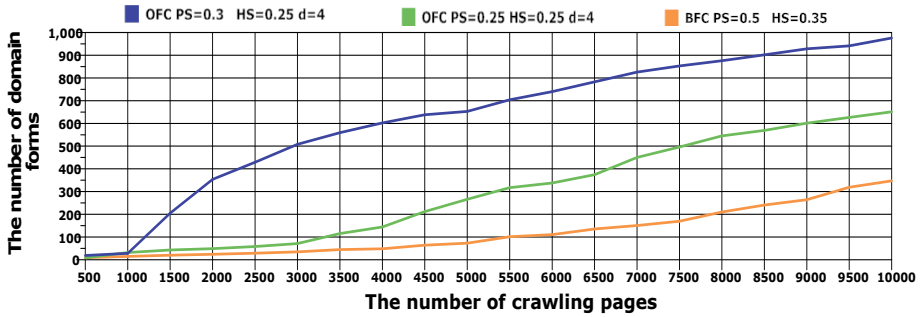


Fig. 12 The number of crawling domain forms for OFC and BFC. From the results of WFF, when PS=0.25, OFC will mistake some irrelevant pages relevant, in this way, it will crawl some useless pages. Therefore, the number of crawling domain forms for PS=0.3 are more than PS=0.25. Compared with BFC, OFC can obtain more Domain-Specific forms than BFC, because that BFC does not consider the page depth, when BFC obtain a page whose page similarity is more than threshold, it will parse the page, however, 94% of the searchable form depth is less than 3. Therefore, BFC has crawled a large number of pages without domain forms.

5. Conclusion

In this paper, we have presented a framework WFF for identifying Deep Web entries based on ontology and focused crawling automatically. Our approach composes three classifiers by partitioning the process into three modules: WPC, FSC and FCC. In the future work, we will conduct further research to improve our work in the following ways: Firstly, we will enrich the ontology, because that the classification accuracy to a large extent depends on the complete ontology knowledge base. Secondly, we will study an effective way of analyzing the hyperlinks in the visited pages to filter the irrelevant pages more efficiently. Finally, we will explore the more effective method to improve the classification accuracy in more depth.

Acknowledgment. This work is supported by the National Natural Science Foundation of China under Grant No.60973040; the National Natural Science Foundation of China under Grant No.60903098; the Science and Technology Development Program of Jilin Province of China under Grant No. 20070533; the Specialized Research Foundation for the Doctoral Program of Higher Education of China under Grant No.200801830021; the Basic Scientific Research Foundation for the Interdisciplinary Research and Innovation Project of Jilin University under Grant No.450060445161;.the Basic Scientific Research Foundation for Young Teachers Innovation Project of Jilin University under Grant No.450060441075.

References

1. Denis Shestakov, Sourav S. Bhowmick, Ee-Peng Lim: DEQUE: Querying the Deep Web. Data & Knowledge Engineering, Vol. 52, 273–311. (2005)
2. Jianguo Lu, Dingding Li: Estimating Deep Web Data Source Size by Capture–Recapture Method. Journal of Information Retrieval, 70-95. (2010)
3. Ritu Khare, Yuan An, Il-Yeol Song: Understanding Deep Web Search Interfaces: A Survey. SIGMOD, Vol. 39, No. 1, 33-40. (2010)
4. Y.Ru and E.Horowitz: Indexing the Invisible Web: A Survey. Online Information Review, Vol. 29, No. 3, 249-265. (2005)
5. Barbosa L and Freire J: Searching for Hidden-Web Databases. In Proceedings of WebDB, 1-6. (2005)
6. Barbosa L, Freire J: Combining Classifiers to Identify Online Databases. In Proceedings of the World Wide Web Conference(WWW), 431-440. (2007)
7. Barbosa L, Freire J: An Adaptive Crawler for Locating Hidden-Web Entry Points. In Proceedings of the World Wide Web Conference(WWW), 441-450. (2007)
8. Luciano Barbosa, Hoa Nguyen, Thanh Nguyen: Creating and Exploring Web Form Repositories. SIGMOD, 1175-1177. (2010)
9. Manuel Alvarez, Juan Raposo, Alberto Pan, Fdel Cacheda, Victor Carneiro: Deep-Bot: A Focused Crawler for Accessing Hidden Web Content. In Proceedings of DEECS, 18-25. (2007)
10. Hui Wang, Yanwei Liu, Wanli Zuo: Using Classifiers to Find Domain-Specific Online Databases Automatically. Journal of Software, Vol. 19, No. 2, 246-256. (2008)
11. Li Yingjun, Nie Tiezheng, Shen Derong, Yu Ge: Domain-oriented Deep Web Data Sources Discovery and Identification. In Proceedings of Asia Pacific Web Conference, 464-467. (2010)
12. Pengyi Zhang, Yan Qu, Chen Huang, Paul T.Jaeger, John Wells, W.Scott Hayes, James E.Hayes Xin Jin: Collaborative Identification and Annotation of Government Deep Web Resources: A Hybrid Approach. 285-286. (2010)
13. Luis Gravano, Panagiotis G.Ipeirotis: QProber: A System for Automatic Classification of Hidden Web Databases. ACM Transactions on Information Systems, Vol. 21, No. 1, 1-41. (2003)
14. Panagiotis G.Ipeirotis, Luis Gravano: Classification-Aware Hidden-Web Text Database Selection. ACM Transactions on Informaion Systems, Vol. 26, No. 2. (2008)
15. Victor Z.Liu, Richard C.Luo, Junghoo Cho, Wesley W. Chu: D_Pro: A Probabilistic Approach for Hidden Web Database Selection Using Dynamic Probing. ICDE, 1-12. (2004)
16. Lu Jiang, Zhaohui Wu, Qinghua Zheng and Jun Liu: Learning Deep Web Crawling with Diverse Features. IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 572-575. (2009)
17. Lu Jiang, Zhaohui Wu, Qian Feng, Jun Liu, and Qinghua Zheng: Efficient Deep Web Crawling Using Reinforcement Learning. PAKDD, 428–439. (2010)
18. Pierre Senellart, Avin Mittal, Daniel Muschick, Remi Gilleron, Marc Tommasi: Automatic Wrapper Induction from Hidden-web Sources with Domain Knowledge. WIDM, 9-16. (2008)
19. Ping Wu, Ji-Rong Wen, Huan Liu, Wei-Ying Ma: Query Selection Techniques for Efficient Crawling of Structured Web Sources. In Proceedings of ICDE, pp47-57. (2006)

20. Elena Simperl: Reusing ontologies on the Semantic Web: A feasibility study. Data & Knowledge Engineering, Vol. 68, 905–925. (2009)
21. Weifeng Su, Jiying Wang, Frederick H.Lochovsky: ODE: Ontology-assisted Data Extraction. ACM Transactions on Database Systems, Vol. 34, No. 2, 1-35. (2009)
22. Matthew Horridge, Bijan Parsia, Ulrike Sattler: Explanation of OWL Entailments in Protege4. In Proceedings of International Semantic Web Conference. (2008)
23. Gechao Lu, Wanli Zuo, Aiqi Zhang, Ying Wang, Wenyan Ji: Ontology-Based Focused Crawler. Journal of Information&Computational Science, 577-584. (2010)
24. K. C.-C. Chang, B. He, C. Li, M. Patel, and Z. Zhang: Structured Databases on the Web: Observations and Implications. SIGMOD Record, Vol. 33, No. 3, 61-70. (2004)
25. Barbosa, L., Freire, J.: Searching for Hidden-web Databases. In: Eighth Intl. workshop on the web and Databases. (2005)
26. Tom M. Mitchell, McGraw Hill. Machine Learning. (1997)
27. S. Sivakumari, R. Praveena Priyadarsini, P. Amudha: Accuracy Evaluation of C4.5 and Naive Bayes Classifiers Using Attribute Ranking Method. International Journal of Computational Intelligence Systems, Vol. 2, No. 1, 60-68. (2009)
28. Quinlan, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers. (1993)
29. http://en.wikipedia.org/wiki/C4.5_algorithm
30. Bing Liu. Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Springer. (2007)
31. Huilan Zhao: Research on Deep Web Sources Classification Technology. In Proceedings of the 2nd International Conference on Future Computer and Communication, 169-171. (2010)
32. Hexiang Xu, Xiulan Hao, Shuyun Wang, Yunfa Hu: A Method of Deep Web Classification. In Proceedings of Machine Learning and Cybernetics, 4009-4014. (2007)
33. Lau A, Tsui E, Lee W.B: An Ontology-based Similarity Measurement for Problem-based Case Reasoning. Expert Systems with Applications, Vol. 36, No. 3, 6574-6579. (2009)
34. S.Chakrabartim, M.van den Berg and B.Dom: Focused Crawling: A New Approach to Topic-specific Web Resource Discovery. Computer Networks, Vol. 31, No. 11-16, 1623-1640. (1999)

Ying Wang was born in 1981. She is a lecturer at the Jilin University and a CCF member. She received her Ph.D. degree from Jilin University. Her research area is Web Information Mining, Ontology and Web search engine.

Huilai Li was born in 1962. He is a professor and doctoral supervisor at the Jilin University. His research areas are partial Differential Equations.

Wanli Zuo was born in 1957. He is a professor and doctoral supervisor at the Jilin University and a CCF senior member. His research areas are database, data mining and Web search engine.

Fengling He was born in 1962. He is a professor at the Jilin University and a CCF senior member. His research areas are database, data mining and Web search engine.

Xin Wang was born in 1981. He is a lecturer at the Changchun Institute of Technology. His research area is Web Information Mining and Ontology.

Kerui Chen was born in 1983. She received her Ph.D. degree from Jilin University. Her research area is Web Information Mining, Ontology and Web search engine.

Received: March 22, 2010; Accepted: January 13, 2011.

DOI: 10.2298/CSIS100327029R

Study of Privacy-preserving Framework for Cloud Storage

Huang RuWei[1,2], Gui XiaoLin[1], Yu Si[1], Zhuang Wei[1]

[1]Department of Electronics and Information Engineering,
Xi'an Jiaotong University, 710049, Xi'an, China
ruweih@126.com
xlgui@mail.xjtu.edu.cn
yusiing@126.com
zhuang2978002@126.com
[2]School of Computer, Electronics and Information,
GuangXi University, 530004, NanNing, China

Abstract. In order to implement a privacy-preserving, efficient and secure data storage and access environment of cloud storage, the following problems must be considered: data index structure, generation and management of keys, data retrieval, treatments of change of users' access right and dynamic operations on data, and interactions among participants. To solve those problems, the interactive protocol among participants is introduced, an extirpation-based key derivation algorithm (EKDA) is designed to manage the keys, a double hashed and weighted Bloom Filter (DWBF) is proposed to retrieve the encrypted keywords, which are combined with lazy revocation, multi-tree structure, asymmetric and symmetric encryptions, to form a privacy-preserving, efficient and secure framework for cloud storage. The experiment and security analysis show that EKDA can reduce the communication and storage overheads efficiently, DWBF supports ciphertext retrieval and can reduce communication, storage and computation overhead as well, and the proposed framework is privacy-preserving while supporting data access efficiently.

Keywords: cloud storage, key derivation, Bloom Filter, privacy security, encrypted keyword retrieval.

1. Introduction

Cloud storage provides on-demand, scalable and QoS guaranteed storage resource, and users can operate their data anytime and anywhere. Facing the powerful and appealing advantages of cloud storage, however, a lot of people and companies are hesitant to put their data in cloud. The main reason is that people and companies are afraid of loss of control on their data. Many vocal consultants, including Gartner, have issued warning on the privacy threats in cloud storage [1]. And there are some incidents of data leakage and loss

which verify people's fears: Google's Docs was visited by unauthorized users because of a software bug in 2009, which caused data leakage [2]; a cloud storage-provider named MediaMax went out of business in 2008 after losing 45% of client data due to an error of a system administrator [3]; criminals targeted the main cloud service provider Salesforce.com, and succeeded in stealing customer emails and addresses using a phishing attack in 2007 [4]. Therefore, to be sustainable, in-depth development, cloud storage must address the privacy concern, efficient and secure data storage and access.

1.1. Related Work

There have been many works on outsourced storage. Josh [5] developed a privacy-preserving electronic health record system. Based on symmetric and asymmetric encryption, it designed two key derivation schemes and compared the advantages and disadvantages of both. But it didn't consider the effects of change of user access right and the dynamic operations of data which would influence the effectiveness of key derivation greatly according to the analysis of the following sections. Brian [6] formalized a model called PDAS for preserving privacy and integrity of aggregate query results. PDAS supported privacy protection by dividing the owner's database into n sections and sending a section to a service provider. Any k ($k \leqq n$) of them can cooperate to recover the entire database, but any smaller group cannot. PDAS didn't encrypt the data, so the service provider can get some information from partial data though it can't get the whole database. And it demanded several service providers to cooperate, which is unrealistic. Wang [7] proposed a scheme to access outsourced data securely and efficiently. It built data index by binary tree, generated and managed keys by key derivation, dealt with the dynamics of access right and data by over-encryption and/or lazy revocation. Its shortcomings include that binary tree structure couldn't reflect the logical relation fully by which owner organizes his data; the scheme of changing user's access right would make other user whose access right doesn't change to update certificate, which will bring additional communication overhead; the scheme of updating data needs to store a control block on service provider, which will occupy additional storage resource and it is uneconomical; it didn't consider how the dynamics of access right and data influences the effectiveness of key derivation; it didn't think about the collusive attacks in which revoked users cooperate with a service provider.

All of the above researches didn't support ciphertext-based retrieval, namely service provider can be an agent of data owner to retrieve the owner's encrypted data according to the user's encrypted query, which protects the privacy of owner and user well. As service provider undertakes the jobs of retrieving data, owner can be relieved from data management, which reflects the advantage of cloud storage. There have been some works on this subject. Liu [8] proposed symmetric encryption-based ciphertext retrieval method. Song [9] proposed an asymmetric encryption-based ciphertext retrieval

scheme. Yasuhiro [10] proposed a scheme based on Bloom Filter to disclose data which match a Boolean query. D.Bonech [11] developed a public key encryption method which enables Alice to provide a key to the gateway that enables the gateway to test whether a keyword is in the email without learning anything else about the email. S.Bellovin [12] proposed a search scheme based on Bloom Filters and group cipher to support ciphertext search. Wong [13] designed an asymmetric scalar-product-preserving encryption to conduct the k-nearest neighbor (KNN) computation on an encrypted database. The above works have some disadvantages: (1) don't support data sharing; (2) have heavy running overheads; (3) sometimes need a semi-trusted third party to transform query or data. Obviously, those researches can't satisfy the demands of cloud storage.

Through the above analysis, we can see that a privacy-preserving, efficient and secure cloud storage framework is needed urgently, which should resolve the following problems: data index structure, generation and management of keys, encrypted keywords retrieval, change of users' access right and dynamic operations on data, and interactions among participants.

1.2. Related Definitions

Definition1 (Key Derivation[14-15]): Data owner organizes his data in tree structure, and chooses a random key as root key, then produces sub key by the following formula: $key_{child}=hash(key_{parent}||child\_number)$, where hash() is a public hash function, and the owner only needs to store the root key. When a user asks for the access authorization, the owner will return a minimum key group from which all authorized files' keys can be derived and other unauthorized files' keys can't.

Definition2 (Bloom Filter[16]): Bloom Filter is a probabilistic data structure to test whether an element is a member of a set. A Bloom Filter is represented by an m-bit array. There are also *k* independent hash functions $h_1,h_2,\ldots, h_k$, which produce outputs that are distributed uniformly over the range [1…m].

A set $S=\{e_1, e_2,\ldots,e_n\}$ is expressed by an m-bit array, which can be realized by the following two steps:

(1) Insertion: Initially, all bits in the bit array are set to 0. To add an element e_i to a Filter, independent hash functions of the element are calculated and array bits at position $h_1(e_i)$, $h_2(e_i)$, …, $h_k(e_i)$ are set to 1. If there are *n* elements $e_1, e_2,\ldots, e_n$ in *S*, this insertion is repeated for each of the elements.

(2) Query: To determine if an element *e* belongs to the set *S*, the bits at positions $h_1(e)$, $h_2(e)$, …, $h_k(e)$ are checked. If any selected bit is 0, *e* is definitely not a member of *S*. On the other hand, if all the checked bits are 1, then *e* is considered as a member of the set.

Since Bloom Filter is a probabilistic data structure, it always has a possibility of false positive, in which *e* appears to be in *S* but actually is no. There are three key parameters which can affect the false positive rate: the number of hash functions *k*, the size of the bit array *m* and the number of keywords *n*. We will compute the false positive rate *p* by formula (1):

$$p = \left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^{k} \approx \left(1 - e^{kn/m}\right)^{k} \quad (1)$$

Formula (1) is minimized when k=(m/n)*ln2, in which case it becomes:

$$p = \left(\frac{1}{2}\right)^{k} = (0.6185)^{m/n} \quad (2)$$

Suppose the false positive rate is less than 0.01% , then k is set to more than 14 and m should be more than 20*n. The false positive rates are shown to be tunable by careful selection of parameters.

The remainder of the paper is organized as follows. Section 2 builds the cloud storage framework and designs the interaction protocol among participants. In section 3, several key issues in the framework are discussed. In section 4, we analyze the performance of the key techniques in the framework. Finally, section 5 concludes the paper and discusses future extensions.

2. The Privacy-preserving Cloud Storage Framework

Figure 1 reflects the functional modules of data owner, user and storage service provider and the interactions among them. The dashed lines show the functional or structural correspondence of the connected parts. The interaction protocol includes five steps as following:

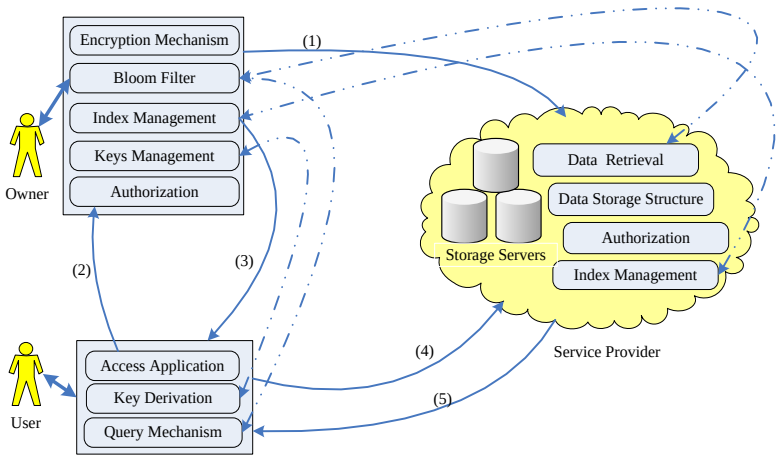


Fig. 1. The privacy-preserving cloud storage framework

(1) Owner(O) chooses a root key key_{root} for file encryption by symmetric encryption algorithm E(), a pair of keys (k_{pub}, k_{pri}) for keywords encryption of

file by asymmetric encryption algorithm AE(). Before $file_i$ is sent to Service Provider(S), owner generates the key k_i of $file_i$ by key derivation algorithm and encrypts $file_i$. Then he encrypts keywords $\{kw_1,kw_2,\ldots,kw_n\}$ by k_{pub} and generates a Bloom Filter BF_i. At last, he sends encrypted files to service provider as following:

$MSG_{OS}=\{O,S, E_{kos}(O, S, E_{k1}(file_1)||BF_1||\ldots||E_{ki}(file_i)||BF_i, t_{modified}, MAC)\}$

And k_{os} is the symmetric key between owner and service provider, $t_{modified}$ reflects the time of last update of the file, MAC(Message Authentication Code) is used to verify the integrity of message.

(2) User(U) requests access authorization from owner. And k_{uo} indicates the symmetric key between user and owner, requestId is the serial number of request:

$MSG_{UO}=\{U, O, E_{kuo}(U, O, requestId, MAC)\}$

(3) Owner verifies user's identity firstly, and searches on access control list to determine the files which can be accessed by user, then sends the minimum key group key_{min} of those files and the certificate(cert) to user. And k_{os} is the symmetric key between owner and service provider, t_{cert} indicates when the certificate is generated, and AR records the update times of the user's access right:

$cert=\{E_{kos}(U, number_{min}, t_{cert}, AR, MAC)\}$

$MSG_{OU}=\{O, U, E_{kuo}(O, U, requestId, number_{min}, key_{min}, cert, MAC)\}$

(4) User sends the certificate to service provider and asks for some files which contain the keyword *kw*. AE() indicates the asymmetric encryption algorithm which is used to encrypt keywords by owner, and k_{pub} is the public key of owner:

$MSG_{US}= \{U, S, O, requestId, AE_{kpub}(kw), cert\}$

(5) Service provider tests the certificate. If it is legitimate, service provider returns those requested files. $E_{ki}(file_i)$ is the file which is encrypted by owner, and service provider never decrypts it.

$MSG_{SU}=\{S, U, requestId, E_{ki}(file_i)||\ldots||E_{kt}(file_t), MAC\}$

User gets the encrypted files, computes the keys of the files from key_{min} by key derivation algorithm, and then decrypts the files. Of course, the files will not be encrypted if they needn't be kept secret.

3. Several Key Issues in Framework

3.1. Index Structure based on Multi-tree

Owner organizes his files in accordance with some logical relations. For reflecting the logical relations, the framework constructs the file index by multi-tree. When those files are going to be stored in the servers of service provider, the client software of owner will generate multi-tree index automatically according to their logical relation, as shown in Fig 2.

In such an index, only leaf nodes correspond to files, and non-leaf nodes represent folders or categories of files. Owner encrypts the content and name of a file and changes its' name as file_number$$E_{kfile\_number}$(file_name), for example 1_2_1$$E_{k1_2_1}$(Diary), before he sends the file to service provider. The pretreatment prevents service provider from knowing the content and name of the file, which protects the owner's privacy. The service provider will construct an index for every owner according the files' numbers, which can accelerate search on data.

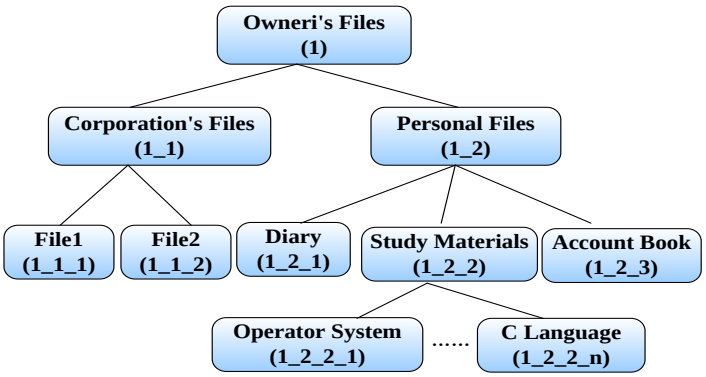


Fig. 2. The index structure of owner's files

3.2. The Extirpation-based Key Derivation Algorithm—EKDA

To have a flexible and fine-grained access control, every file has a unique key. The framework uses symmetric encryption algorithm AES to reduce the burden of encryption and decryption. But how to manage the numerous keys? Key derivation can be used to solve the problem. Owner chooses a random 128-bit key as root key, then produces subkey by the following formula: k_{child}=hash(k_{parent}||child_number), and hash() is a public hash function, for example, SHA-1. Owner only needs to store the root key, which is not only convenient to key management, but also saves the owner's storage space. When a user asks for the access authorization, the owner will return the minimum key group from which all authorized files' keys can be derived and other unauthorized files' keys can't. Key derivation can reduce the communication overhead of participants effectively.

But the effectiveness of key derivation will be harmed in some case: when the access right of a user is changed, owner must use a new key to encrypt the files if owner don't want the user to access the files again. The new key can't be computed by k_{child}=hash(k_{parent}||child_number), and the framework generates the new key by choosing a random 128-bit number. When there are a lot of files using new key or every penultimate level directory has a file

using new key, the effect of key derivation is the same as the situation where key derivation is not used, namely owner must return N keys if there are N authorized files.

To solve this problem, we design an extirpation-based key derivation algorithm(EKDA): owner labels the node with “updated” which will use a new key because of the change of user access right in the index tree, and creates a new node in the update tree. The new node has the same number with the original node and has a new key. The course is shown in Fig 3. When the node needs to update again, it can change the key of the node in the update tree. When user requests access authorization, owner will compute the minimum key group by EKDA. The algorithm is as following:

```
public String EKDA(Node[] nodes,int types)
 { if(types==0){  // node[0] is updated in first time
     nodes[0].setUpdated();
     Node updatedNode=new
                   Node(nodes[0].number,keyRandom());
     updateTree.addNode(updatedNode);
     String key=updateNode.getKey();
     encrypt(file,key);
   }else if(types==1){
      // node[0] is updated in non-first time
     String key=Nodes[0].getUpdatedNode().createNewKey();
     encrypt(file, key);
   }else{   // compute the minimum key group
     String key_min="";
     Node parentNode=null;
     for(int i=0;i<nodes.length;i++){
       if(nodes[i].updated==1)

key_min=key_min+nodes[i].getUpdateNode().getKey();
       else{
         parentNode=findParentNode(i,nodes);
         if(parentNode!=null){
           key_min=key_min+parentNode.getKey();
           Node[] newNodes=nextNodes(nodes,parentNode);
           String s=extirpated_keyderivation(newNodes, 3);
           key_min=key_min+s;
         }else key_min=key_min+nodes[i].getKey();
  }}}
  return key_min;
}
```

Here is an example. When owner updates the key of file 1_2_2, he will get the file from service provider and decrypt it by the old key firstly. He asks service provider to delete the file and mark the node 1_2_2 with “updated”. Then he encrypts the file’s content and name with new key of the node in the update tree, and sends the encrypted file to the service provider. When an authorized user can access the files under the folder 1_2, the owner searches the index tree and returns the minimum key group which includes key1_2 and

key1_2_2. From the effect of the algorithm, node 1_2_2 seems to be extirpated from the index tree. The algorithm can reduce the number of returned key effectively.

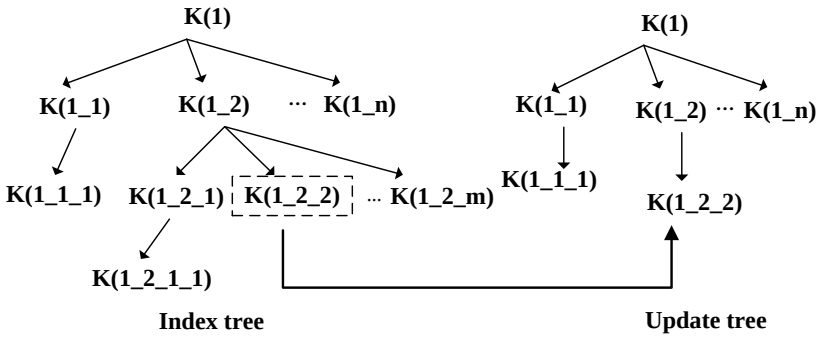


Fig. 3. The correspondence between index tree and update tree

3.3. The Double Hashed and Weighted Bloom Filter——DWBF

Bloom Filter is usually used to store a great deal of elements' information, for example, ten thousand elements. But in our framework, a file has a Bloom Filter which is used to record the keywords of the file and help ciphertext retrieving. And the number of a file's keywords is always less than 50. So the Bloom Filter in our framework is a small Bloom Filter.

Although Bloom Filter has good performance to reduce communication and storage overhead, it brings a lot of computation overhead. For example, when there are *z* keywords in a file, and there are *k* independent hash functions used in the Bloom Filter, the computation overhead to produce a file's Bloom Filter is *z*k* hash calculations; when the service provider queries a files, he need to do *k* hash calculations. So if there are large numbers of files, the computation overhead of owners and service providers is tremendous. So how to reduce the computation overhead of Bloom Filter is a key problem of ciphertext retrieval scheme in cloud.

In addition to reduce computation overhead of Bloom Filter, how to reduce the cost of false positive of Bloom Filter is another key problem. For example, when the situation of false positive occurs, 10M file and 256k file will have different effects on the communication overhead and the user's satisfaction. So, we should try to reduce the cost of false positive, and at the same time, reduce the computation overhead.

(1) Reduction in computation overhead

The computation overhead of Bloom Filter is mainly used to compute hash functions. So if we can reduce the number of hash functions in a Bloom Filter and can get the same false positive rate with the standard Bloom Filter, the

computation overhead can be reduced. There are some researches on the problem. P.C.Dillinger and D.Knuth [17-19] introduced the double hashing technologies, which used two hash functions $h_1(x)$ and $h_2(x)$ to simulate more than two hash functions of the form $g_i(x)= h_1(x)+i*h_2(x)+f(i)$. A.Kirsch [21] evaluated the above methods and got the following results by theoretical analysis, as shown in formula (3):

$$\lim_{n\to\infty} p = (1-e^{-k/c})^k \quad (3)$$

which *p* is the false positive rate of Bloom Filter, *c=m/n*, and *m* is the size of the bit array and *n* is the number of keywords. That is to say, when the number of keywords tends to infinity, the false positive rate of Bloom Filter with two hashing technique is equal to that of standard Bloom Filter. But in our framework, the number of a file's keywords is less than 50, so we want to know whether two hashing techniques are useful when the number of elements is small? We designed the following experiments: we choose the following specific Bloom Filters: the standard Bloom Filter(SBF), the double hashed Bloom Filter(DBF) which $g_i(x)=h_1(x)+i*h_2(x)$, DBF2 which $g_i(x)= h_1(x)+i*h_2(x)+i^2$, DBF3 which $g_i(x)= h_1(x)+i*h_2(x)+i^3$, and DBF4 which $g_i(x)= h_1(x)+i*h_2(x)+i^4$. Then we compute value of $k\in\{\lceil cLn2\rceil, \lfloor cLn2\rfloor\}$ that minimizes $p=(1-\exp(-k/c))^k$. Next, for each of the Bloom Filters under consideration, repeat the following procedure 1000 times: instantiate the Filter with the specified values of *n*, *c* and *k*, populate the Filter with n ciphertext of keywords, and then query $\lceil 10/p\rceil$ elements not in *S*, recording the false positive rate of those Bloom Filters. The result is showed in Fig 4.

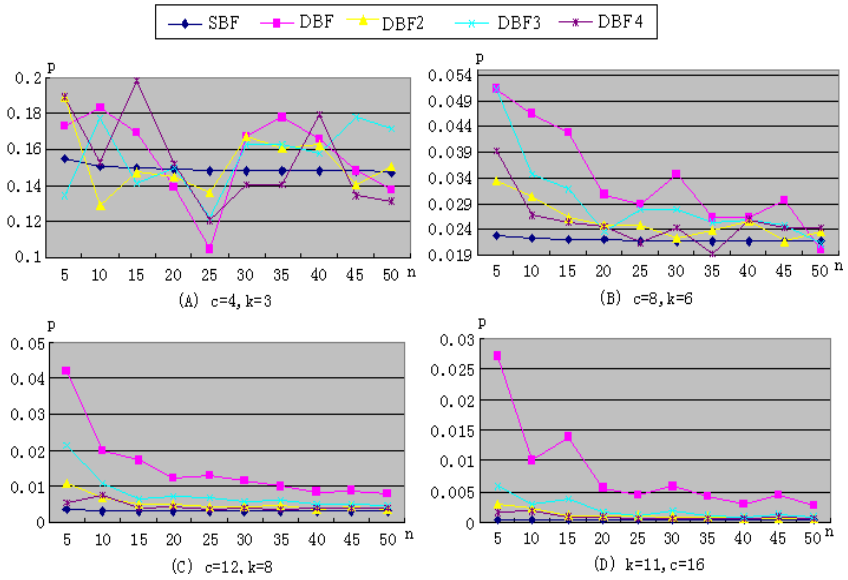


Fig. 4. The false positive rate of several Bloom Filters

From Fig 4, we can find when k≥8, the false positive rate of DBF4 is closest to that of standard Bloom Filter. So we can replace the standard Bloom Filter with DBF4 (k≥8). If there are k hash functions used in SBF, the computation overhead of DBF4 is k/2 times as small as that of SBF.

(2) Reduction in cost of false positive

There are some works on the subject. Bruck [21] incorporated the information on the query frequencies and the membership likelihood of the elements into its optimal design; Xie [22] dealt with different elements in a data set depending on their query invalidation cost by clustering elements into different baskets. Those works are suitable for the owner to manage his own files. But in our framework, the situation is different. The Owner outsources his files and the service provider couldn't know how many files will be stored in his servers, and how big the files are, and how often the files are visited before the files are stored. So their methods couldn't be fit for our framework.

So we design a simple scheme based on file's size. Of course, query frequency can be used if the owner knows. But because of data sharing, the owner may have no idea about it. Suppose service provider divides all files into three groups according to their sizes, for example, the files whose size are smaller than 1M belong to the first group, the files whose size is bigger than 1M and smaller than 64M belong to second group, and the file whose size is bigger than 64M belong to the third group. And the relation of the false positive rate of the three groups is: $p_{group1}= 2*p_{group2}= 4*p_{group3}$. Suppose k_i means how many hash functions the Bloom Filter of $group_i$ uses, we can get the result: $k_2=k_1+1$, $k_3=k_2+1=k_1+2$.

(3) The formal description of DWBF

Now, we give the formal description of DWBF(Double Hashed and Weighted Bloom Filter) as following:

$$DWBF=\{\{\{kw_{11},\ldots,kw_{1n}\},\{kw_{21},..,\ kw_{2m}\},\ldots,\{\ kw_{q1},\ldots,\ kw_{qz}\}\},$$
$$\{w_1,w_2,..,w_q\},\{k_1,k_2,\ldots,k_q\},\ \{sw_1,sw_2,\ldots sw_j\}$$
$$g_i(x)= h_1(x)+t*h_2(x)+t^4,\ t\in Z \text{ and } t\in[0,k_i)\}$$
$$(n,m,z,q,j\in N,\ k_i\in\{k_1,k_2,\ldots,k_q\} \text{ and } k_{j+1}=k_j+1(j\geq 1)).$$

The definition means there are *q* files, and every file has some keywords, w_i is the size of $file_i$, and k_i is the number of hash functions used in the Bloom filter of $file_i$; sw_i is the weight standard to divide files into j+1 groups according to file's size, for example, {1M,64M} means there are three groups, whose range is size≤1M, 1M<size≤64M, and size>64M respectively.

3.4. Change of Access Right

Service provider builds an access right updating list updateAR[Owner_id] for every owner firstly. And the node in list has two properties: node.id is the number of user, and node.times indicates how many times the access right of the user was updated. After $Owner_i$ updates the access right of $User_j$, he sends the update massage to service provider with the number of $User_j$.

Service provider receives the massage and searches the list updateAR[i]. If there is a node with node.id=j, then node.times++; otherwise service provider inserts a new node into updateAR[i] and set node.id=j and node.times=1. When $User_j$ requests files, service provider checks whether there is a node with node.id=j in updateAR[i]. If there is not such a node, service provider returns the files; if there is such a node, service provider will check whether node.times is equal to cert.AR. If node.times is equal to cert.AR, he will return the files; otherwise he will refuse to return the files and remind the $User_j$ that his certification has expired. The above operations prevent revoked user from getting files from service provider.

Of course, a revoked user can steal files when the files are transmitted. There are two methods to solve the problem: one is over-encryption [23] and the other is lazy revocation [24]. Over-encryption asks the service provider to encrypt the files before they are transmitted, which can prevent revoked user getting the files, but not all service providers are willing to provide such a service and encrypting a batch of files increases the economic burden of owner. Lazy revocation doesn't need owner and service provider to do anything before the file is updated because the stolen file is the same as the file which the revoked user had authorization to access. The framework adopts lazy revocation.

3.5. Dynamic Data Operations

Owner has three dynamic operations on data: storage, deletion and update. When owner wants to store a new file, he will find a new number from the index tree according to the logical relation and compute the key by $k_{child}=hash(k_{parent}||child\_number)$, and then encrypt the file and send it to service provider. When owner wants to delete a file, he will send a delete message to service provider to delete the file, then mark the node of the file in the index tree with "deleted". When there is a new file which wants to use the number of the deleted file, it will be treated as an updated file.

When the file is updated, the key is valid if there is not a revoked user who could access the file before. Otherwise we need to do the following operations: owner marks the node of the file in index tree with "updated", and inserts a new node with same number and new key into update tree. Then he encrypts the content and name of the file with new key, and sends the encrypted file to service provider. Suppose $t_{modified}$ indicates when the file was modified latest and t_{cert} indicates when the user's certificate was created. When a user requests the file, service provider compares $t_{modified}$ and t_{cert}, cert.AR and node.times of node whose node.id is equal to the user's number in updateAR[owner_id]. If $t_{modified}>t_{cert}$ and cert.AR==node.time, the user is an authorized user whose key is old, so service provider will return the file and remind him to get a new key; if $t_{modified}\leqq t_{cert}$ and cert.AR==node.time, the user is an authorized user who's key is new, so service provider will return the file; if cert.AR<node.time, the user is an revoked user, so service provider will refuse to return the file to him.

4. Performance Evaluation

Our research group is designing and developing a campus-level cloud computing platform named “Qing Cloud”. Based on the above framework, we developed a system of cloud storage by Java. Now we will evaluate the feasibility of the framework by analyzing the effectiveness of EKDA and DWBF, the run-time overhead of the system and privacy security.

4.1. Effectiveness of EKDA

To reflect the real cloud storage environment, the experiment simulates the interactions among multiple users, multiple owners and multiple service providers in Qing Cloud. User requests files, and owner changes users’ access right and updates files randomly. Owner stores one hundred files in different organization structures in servers of service provider. Suppose the size of minimum key group of EKDA is *size1*, and the size of minimum key group of common key derivation is *size2*. By computing *size1*/*size2*, the effectiveness of extirpation-based key derivation can be verified, which is showed as Fig 5.

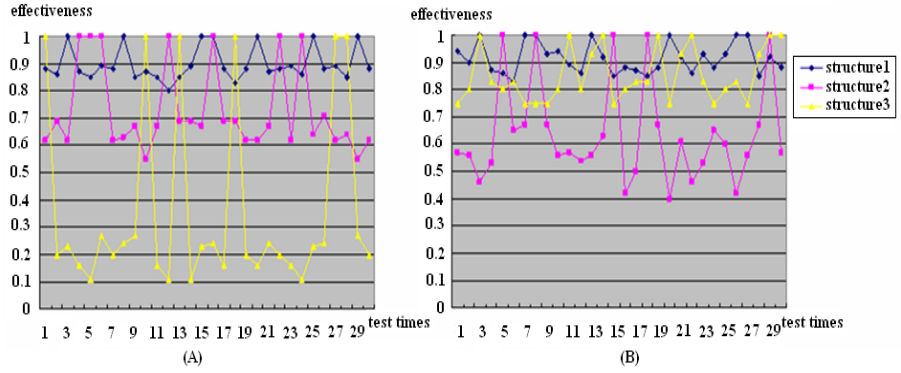


Fig. 5. The effectiveness of EKDA

Figure 5(A) shows the effectiveness when updating the same file in three different file organization structures; figure 5(B) shows the effectiveness when updating another file in the above three structures. From figure 5(A) and 5(B), we can draw the following conclusions: (1) EKDA is very effective because size1/size2≦1; (2) the organization structure of files has a direct influence on the effectiveness of the algorithm; (3) the position of the updated file has a direct influence on the effectiveness of the algorithm; (4) when a file organization structure is fixed, except those points whose value is 1, the points always surround a value. The reason is that the effectiveness of the

algorithm is *(1+m)/n* if there are *n* files in a folder which has *m* updated files. So the effectiveness will fluctuate around *(1+m)/n*.

4.2. Effectiveness of DWBF

We will analyze the performance of Bloom filter from computation, communication, storage overheads and the false positive rate.

(1) Reduction in Communication overhead and false positive rate

We will compare the communication overhead and false positive rate of DWBF with those of standard Bloom filter(SBF). Suppose there are two weight standards to divide the files: (i){1M, 64M}; (ii) {256K, 1M, 64M}, and k_1=8 to all owner. There are 100 files whose size ranges from 1k to 1G according to some percentages. For example, if the number of files whose size is smaller than 1M is *n1*, bigger than 1M and smaller than 64M is *n2*, or bigger than 64M is *n3*, 4:4:4 means n1:n2:n3. Every file has ten keywords. And *p1* is the false positive rate of DWBF, *p2* is the false positive rate of SBF, *w1* is the communication overhead of DWBF produced by false positive, and *w2* is the communication overhead of SBF. We repeat the experiment 100 times, and query [$10/p_{min}$] elements not in the keywords of 100 files every time. The result is showed in Fig 6. We can draw the following conclusions: (1) all of ratios are less than 1, so DWBF is good at reducing false positive rate and communication overhead; (2) the more big files there is, the smaller the false positive rate is, which is shown in figure 6(A) and 6(C); (3) the more levels a weight standard has, the smaller the cost of false positive is, which is shown in figure 6(B) and 6(D).

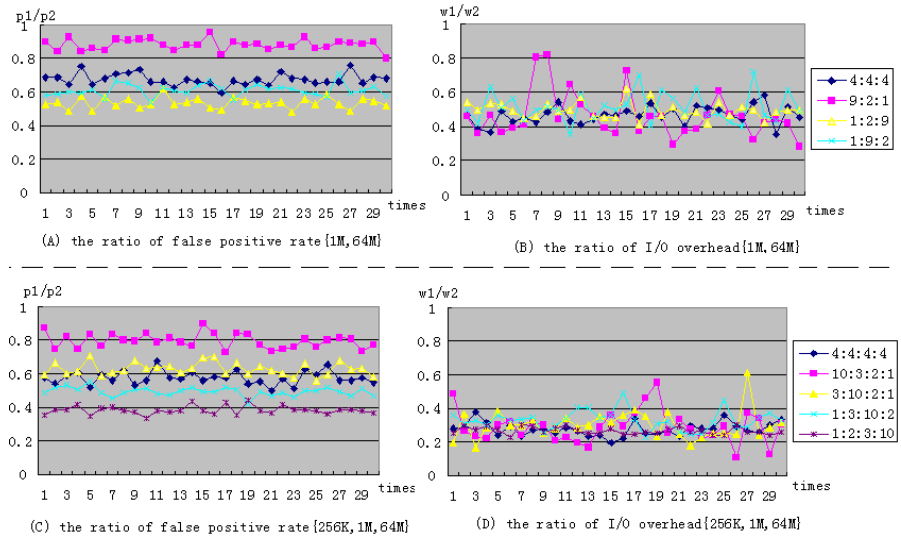


Fig. 6. The performance of DWBF compared with SBF

(2) Overhead of DWBF

In the framework, elliptic curve encryption algorithm(ECC) is used as the asymmetric encryption method which adopts 160-bit key. Owner encrypts keywords of files by ECC, and then transforms the ciphertexts of a file's keywords into a Bloom Filter. Suppose there is a file which has 10, 20, 30, 40, 50 keywords respectively, and the size and the keywords of the file is generated randomly. The result is showed in Table 1.

Table 1. Overhead of DWBF

		Storage and Communication Overhead			Computation Overhead	
num	k	ECC(bit)	DWBF(bit)	DWBF/ECC	Owner(ms)	Service Provider(ms)
10	8	2678	115	0.00043%	0.97	0.1
	9	2682	129	0.00048%	1.01	0.09
	10	2681	144	0.00054%	1.11	0.11
20	8	5367	230	0.00043%	1.97	0.1
	9	5355	259	0.00048%	1.96	0.1
	10	5370	288	0.00054%	1.98	0.1
30	8	8049	346	0.00043%	2.85	0.1
	9	8046	389	0.00048%	2.9	0.1
	10	8044	432	0.00054%	2.93	0.1
40	8	10737	461	0.00043%	3.83	0.1
	9	10733	519	0.00048%	3.89	0.1
	10	10728	577	0.00054%	3.93	0.1
50	8	13411	577	0.00043%	4.71	0.09
	9	13420	649	0.00048%	4.83	0.1
	10	13442	721	0.00054%	5.03	0.09

From Table 1, we can draw the following conclusions: (1) DWBF can reduce the communication and storage overheads greatly; (2) the computation overhead of DWBF is so small and increases so slow that it wouldn't be a burden to owner; (3) DWBF has good query performance because the computation overhead of query keeps around 0.1ms with the increment of keywords.

4.3. Run-time Overhead of the System

Run-time overhead of the whole system is measured from three aspects: communication, computation and storage overhead, as showed in Table 2.

Suppose the amount of files which is authorized to access by $User_j$ is n_j, the amount of files which satisfies $User_j$'s query is s_j; $Owner_i$ has m_i users, the size of encrypted $file_k$ is f_k and the length of its DWBF is bf_k, the high of index tree is h and the nodes in index tree occupies q bits, $Owner_i$ has p files, and the average amount of keywords of every file is g; we adopt 128-bit key, *hash()* indicates the computation overhead of hash function, *E()* and *D()* is the

computation overhead of encrypting and decrypting file by symmetric encryption, *AE()* is the computation overhead of encrypting keyword by asymmetric encryption; *len* is the key amount in minimum key group generated by key derivation. The analysis is shown in Table 2. And (O) indicates that owner undertakes the overhead, the rest may be deduced by analogy.

Our framework reduces run-time overhead immensely by the following measures: (1) storage overhead of owner, communication overhead of number group and key group is reduced greatly by EKDA; (2) According to the framework, file retrieval is done by service provider instead of owner, which relieves the computation overhead of owner; (3) DWBF can reduce the storage, communication and computation overhead as well; (4) To use multi-tree structure, the length of file's serial number is shorter than or equal to the height of the tree, which reduces the communication overhead of number group.

Table 2. Run-time Overhead of the System

Type		Overhead
Communication Overhead	minimum key group	len*128
	minimum number group	len*(h/2)
	file and DWBF	$\sum_{k}^{p}{}_{1}(f_k+bf_k)$
Computation Overhead	ciphertext retrieval(S)	(1/2)*n_j*2*hash( )
	key derivation (O)	n_j*(1/2)*(1+h)*hash( )
	key derivation (U)	s_j*(1/2)*(1+h)*hash( )
	file and keywords encryption(O)	p*(E( )+g*(AE( )+2*hash( )))
Storage Overhead	key(O)	128
	IndexTree(O)	p*h/2*q
	IndexTree(S)	p*h/2*q*m_i
	file and bloom Filter	$\sum_{k}^{p}{}_{1}(f_k+bf_k)$

4.4. Privacy Security

From Fig 1, we can find there are several potential threats to users' privacy: (1) during the course of files transmitting, outside attacker can steal the files by eaves-dropping; (2) inside attacker is easy to steal the files because the files are stored in service provider's servers; (3) a malicious user and a service provider cooperate to steal the owner's files, which is called as collusive attack; (4) when the user queries, service provider may take a peep at the content of query which is the privacy of user.

Aimed at the first attack, attacker can't decrypt the file if he hasn't key. To the second attack, owner encrypts the content and name of the files by symmetric encryption, encrypts keywords of files by asymmetric encryption, and transforms encrypted keywords into BF by hash functions. So the encrypted files and BF are stored in servers of service provider. The

symmetric keys are transmitted from owner to user, the private key of asymmetric keys is only known by owner, and the ciphertexts of keywords are not stored in service provider. So the insider attacker couldn't decrypt the files and keywords.

Against the third attack, owner's certificate limits the scope of files which can be accessed by a user. And certificate is encrypted by the symmetric key K_{os}, which just can be decrypted by owner and service provider, so it can prevent the user from retrieving other files which is out of the scope. Because of the false positive rate of Bloom Filter, service provider may return the files which don't meet the query. But the file is in the scope of authorization, so it won't leak owner' privacy. When service provider is in collusion with malicious users and retrieves files which is out of the authorized scopes, service provider can find the files meeting the query, but he can't decrypt those files because he haven't the keys.

If service provider wants to know the content of user's query, it can only do that by exhaust algorithm because he hasn't the private key of owner. Support there are eighty-five letters of which a filename can be made in alphabet, when there is a five-letter keyword, it spends 30ms to encrypt a string and retrieve Bloom Filter one time by a computer with 1.86GH dual-core CPU and 2GB memory. So, 2.11 years will be spent to find out the five-letter keyword, which is considered as difficult calculation. So the privacy of users can be protected.

From the above analysis, the proposed framework does well in privacy security.

5. Conclusion

In this paper, we propose a privacy-preserving cloud storage framework, which includes an interactive protocol among participants, multi-tree index, extirpation-based key derivation algorithm(EKDA) for key management and double hashed and weighted Bloom Filter-based search on encrypted keyword(DWBF), which are combined with lazy revocation to deal with the changes of users' access right and dynamic operations of data. The framework supports the interactions among multiple users, multiple owners and multiple service providers, but only supports owner-write-user-read. The experiment and security analysis show that EKDA can reduce the communication and storage overheads efficiently, DWBF supports encrypted keywords retrieval and can reduce communication, storage and computation overhead as well, and the proposed framework is privacy-preserving while supporting data access efficiently.

To support privacy protection further, the future works include the following aspects. First, we plan to improve the EKDA to adapt the change of access right better. Second, we are going to study the encryption techniques which support ciphertext computing. Finally, we will integrate the cloud storage system with the virtual machine system which is developed by our research

group, and realize a real cloud environment which supports computation and storage.

Acknowledgments. This work was supported in part by the National Science Foundation of China under Grant No.60873071, the National Science Foundation of China under Grant No.91018011, the National High-Tech Research and Development Plan of China under Grant No.2008AA01Z410, and IBM' Shared University Research Plan.

References

1. Gartner: Assessing the Security Risks of Cloud Computing(ID Number:G0015778 2).(2008)
2. Jessica E. V.: Google Discloses Privacy Glitch(2009). [Online]. Available: http://blogs.wsj.com/digits/2009/03/08/1214/ (current June 2011)
3. Michael K.: MediaMax/The Linkup: When the cloud fails (2008). [Online]. Available: http://blogs.zdnet.com/projectfailures/?p=999(current June 2011)
4. Greenberg, A.: Cloud Computing's Stormy Side(2008).[Online]. Available: http://www.forbes.com/2008/02/17/web-application-cloud-tech-intel-cx_ag_0219 cloud.html(current June 2011)
5. Josh Benaloh, Melissa Chase, Eric Horvitz, Kristin Lauter: Patient controlled encryption: ensuring privacy of electronic medical records. In Proceedings of the 2009 ACM workshop on Cloud computing security. Chicago, IL, USA, 103-114.(2009)
6. Brian Thompson, Stuart Haber, William G. Horne, Tomas Sander, Danfeng Yao: Privacy-Preserving Computation and Verification of Aggregate Queries on Outsourced Databases. In Proceedings of the 9th International Symposium Enhancing Privacy Enhancing Technologies, Seattle, WA, 185-201. (2009)
7. Weichao Wang, Zhiwei Li, Rodney Owens, and Bharat Bhargava: Secure and Efficient Access to Outsourced Data. In Proceedings of the 2009 ACM workshop on Cloud computing security. Chicago, Illinois, USA, 55-66. (2009)
8. Qin Liu, Guojun Wang, Jie Wu: An Efficient Privacy Preserving Keyword Search Scheme in Cloud Computing. In Proceedings of the 2009 International Conference on Computational Science and Engineering. Vancouver, BC, Canada, 715-720. (2009)
9. Dawn Xiaodong Song, PDavid Wagner, PAdrian Perrig. Practical Techniques for Searches on Encrypted Data. In Proceedings of the 2000 IEEE Symposium on Security and Privacy. Berkeley, California, USA, 44-55. (2000)
10. Yasuhiro Ohtaki: Partial Disclosure of Searchable Encrypted Data with Support for Boolean Queries. In Proceedings of the 2008 Third International Conference on Availability, Reliability and Security. BARCELONA, SPAIN, 1083-1090. (2008)
11. D.Bonech, G.D.Crescenzo, R.Ostrovsky, and G.Persiano: Public-key encryption with keyword search. In Proceedings of Eurocrypt 2004. Interlaken, Switzerland, 506-522. (2004)
12. S. Bellovin and W.Cheswick: Privacy-enhanced searches using encrypted Bloom Filters[Online]. Cryptology ePrint Archive, Report 2004/022. Available: http://eprint.iacr.org/2004/022.pdf. (2004)
13. Wai Kit Wong, David Wai-lok Cheung, Ben Kao, Nikos Mamoulis: Secure kNN computation on encrypted databases. In Proceedings of the 35th SIGMOD

international conference on Management of data. Providence, Rhode Island, USA, 139-152. (2009)
14. M.J.Atallah, M.Blanton, N.Fazio, and K.B.Frikken: Dynamic and Effi cient Key Management for Access Hierarchies In Proceedings of the 12th ACM conference on Computer and communications security. Alexandria, VA, USA, 190–202. (2005)
15. RuWei Huang, Si Yu, Wei Zhuang, XiaoLin Gui: Design of Privacy-Preserving Cloud Storage Framework. In Proceedings of the 9th International Conference on Grid and Cloud Computing. Nanjing,China, 128-132. (2010)
16. B.Bloom: Space/time Trade-offs in Hash Coding with Allowable Errors. Communications of the ACM, 13(7), 422-426. (1970)
17. P.C. Dillinger and P. Manolios: Bloom Filter in Probabilistic Verification. In Proceedings of the 5th International Conference on Formal Methods in Computer-Aided Design. Austin, Texas, USA, 367-381. (2004)
18. P.C. Dillinger and P. Manolios: Fast and Accurate Bitstate Verification fro SPIN. In Proceedings of the 11th International SPIN Workshop on Model Checking of Software. Barcelona, Spain, 57-75. (2004)
19. D.Knuth: The Art of Computer Programming, Volume 3: Sorting and Searching. Addison-Wesley. (1973)
20. A.Kirsch and M.Mitzenmacher: Building a Better Bloom Filter. In Proceedings of the 14th Annual European Symposium on Algorithms. Zürich, Switzerland, 1-33. (2006)
21. Jehoshua Bruck, Jie Gao, and Anxiao Jiang. Weighted Bloom Filter. The 2006 IEEE International Symposium on Information Theory. Pasadena, CA, 2304-2308. (2006)
22. Kun Xie, Yinghua Min, Dafang Zhang, Gaogang Xie, and Jigang Wen: Basket Bloom Filters for Menbership Queries. TENCON 2005. Melbourne, Australia, 1-6. (2005)
23. S.D.C.di Vimercati, S.Foresti, S.Jajodia, S.Paraboschi, and P.Samarati: Over-encryption: Management of Access Control Evolution on OutSourced Data. In Proceedings of the 33th international conference on Very large databases. Vienna, Austria, 123-134. (2007)
24. M.Kallahalla, E.Riedel, R.Swaminathan, Q.Wang, and K.Fu. Plutus: Scalable Secure File Sharing on Untrusted Storage. In Proceedings of the FAST '03 Conference on File and Storage Technologies. San Francisco, California, USA, 29-42. (2003)

Huang RuWei received BSc and MSc in Computer Science from GuangXi University in China in 2001 and 2004, respectively. From 2007, she studied as a PhD student in Xi'an Jiaotong University in China. She has been active in Network Computing, Service Computing and Network Security. She has published 3 articles in international and national journals, 3 articles in the proceedings of international conferences.

Gui XiaoLin graduated with a BSc in Computer at Xi'an Jiaotong University of China, and received MSc and PhD in Computer Science from Xi'an Jiaotong University in China in 1993 and 2001, respectively. Since joining Xi'an Jiaotong University in 1988, he has been an active researcher in Network

Computing, Network Security, and Wireless Networks. His recent research covers secure computation of open network systems (include Grid, P2P, Cloud); dynamic trust management theory; development on community network. He has published more than 80 articles in international and national journals. He was the recipient of New Century Excellent Talents in University of China in 2005.

Yu Si received BSc in Computer Science from ShaanXi University of Science and Technology in 2008. From 2008, he began a continuous academic project that involves postgraduate and doctoral study in Xi'an Jiaotong University. He has been active in Cloud Computing and Virtualization security. He has published 1 article in national journal.

Zhuang Wei received BSc in Computer Science from Xi'an Jiaotong University in 2010. From 2010, he studied as a Graduate student in Xi'an Jiaotong University in China. He has been active in Cloud Computing and Virtualization security.

Received: March 27, 2010; Accepted: January 19, 2011.

DOI: 10.2298/CSIS101012030Z

SVM Based Forest Fire Detection Using Static and Dynamic Features

Jianhui Zhao, Zhong Zhang, Shizhong Han, Chengzhang Qu
Zhiyong Yuan, and Dengyi Zhang

Computer School, Wuhan University, Wuhan, Hubei, 430072, PR China
jianhuizhao@whu.edu.cn

Abstract. A novel approach is proposed in this paper for automatic forest fire detection from video. Based on 3D point cloud of the collected sample fire pixels, Gaussian mixture model is built and helps segment some possible flame regions in single image. Then the new specific flame pattern is defined for forest, and three types of fire colors are labeled accordingly. With 11 static features including color distributions, texture parameters and shape roundness, the static SVM classifier is trained and filters the segmented results. Using defined overlapping degree and varying degree, the remained candidate regions are matched among consecutive frames. Subsequently the variations of color, texture, roundness, area, contour are computed, then the average and the mean square deviation of them are obtained. Together with the flickering frequency from temporal wavelet based Fourier descriptors analysis of flame contour, 27 dynamic features are used to train the dynamic SVM classifier, which is applied for final decision. Our approach has been tested with dozens of video clips, and it can detect forest fire while recognize the fire like objects, such as red house, bright light and flying flag. Except for the acceptable accuracy, our detection algorithm performs in real time, which proves its value for computer vision based forest fire surveillance.

Keywords: Forest flame, Color segmentation, Static feature, Shape matching, Dynamic feature, SVM.

1. Introduction

Currently many institutions are trying to develop reliable and efficient methods to forecast the fire disasters, which may induce heavy casualty and property loss as well as serious social impact. The traditional method to detect fire is employing some people as inspectors, but human resource is expensive and such approach has very low efficiency. Fire sensors have already been used as another method to detect the particles generated by smoke or fire, temperature, relative humidity, etc. But they must be placed in the proximity of fire or their detecting range is usually exceeded, and the approach fails to supply the additional information about the process of burning, such as fire

location, size, growing rate, and so on. Fortunately, computer vision based fire detection brings us a new kind of method which can overcome the key deficiencies of the aforementioned methods. However, this new approach still remains immature, and many tough problems exist in its 3 main stages: image segmentation, target tracking and object classification. What's more, one certain detection algorithm cannot work well for all kinds of fire disasters, e.g. tunnel fire, building fire or forest fire.

Moving objects' estimation is often used to segment the possible fire region from video sequence, and two main traditional algorithms include consecutive frames subtraction and background subtraction [1-6]. In the algorithm of consecutive frames subtraction, transient change of image can be detected, but the overlapping region of two consecutive frames can be mistakenly taken as background. In the algorithm of background subtraction, intact target region can be extracted because of the static state of the background image, but the extracted target may be vague and inaccurate if the background image cannot be updated in time. For forest environment, the whole scene does not keep still due to waving trees, changing weather, varying light, moving shadow, shaking camera, and so on.

Therefore, compared with moving estimation, color based segmentation is more suitable for forest fire extraction. Celik et al. [7] described the color features of fire in RGB color space, and decided whether one pixel belongs to the fire region using rules represented by two groups of inequations. Chen et al. [8] tried to segment the fire region from one image in RGB color space based on three deduced decision rules, within which the saturation value of each extracted possible fire pixel needs to be more than one threshold value in order to exclude the other fire-like regions. Celik and Demirel [9] proposed a generic color model for flame pixels' classification in YCbCr color space with several rules, and they used three polynomials to model the region containing 16309070 fire pixels from 1000 sample images in CbCr chrominance plane. Phillips et al. [10] used test data where the fire has been isolated manually to create a color lookup table by creating a Gaussian smoothed color histogram to detect the fire colored pixels, thus their approach based upon training is scene specific with increased accuracy if training sequences are available for specific kinds of fires. Toreyin et al. [11] obtained the fire color distribution from sample images, and represented the 3D point cloud in RGB color space using a mixture of Gaussians, then the pixel with color value inside one of the distributed Gaussian spheres is assumed to be a fire colored pixel. Krstinic et al. [12] compared the lookup table method and the probabilistic model method, and their experiments proved that lookup table classifier achieves the lower performance. Further validations [13,14] have been performed based on fire color based segmentation to improve the extraction accuracy.

Besides color information, dynamic features of fire are significant clues used to distinguish fire from other fire-like objects [15-21]. To identify a fire's growth, size variation of fire area is calculated from some consecutive video frames [17]. If the number of extracted fire pixels is increasing with time and greater than the explicitly defined threshold, a fire alarm will be given. Flickering frequency is an important clue for fire since flames flicker with a

characteristic frequency of around 10 Hz independent of burning material and burner, and there are several approaches to compute fire flickering frequency [18-20]. Zhang et al. [18] thought the height of flame is changeable due to flame flicker and the changing pattern differs from the jamming sources, so the change of flame height is taken as a dynamic feature. Yuan et al. [19] directly utilized the temporal variation of flame contour as clues for detecting whether a pixel is fire or not in video images. The stochastic characteristics of fire motion are estimated by an autoregressive model of changes in Fourier coefficients of the region boundary [20], and temporal changes of the coefficients are used as the signatures of fire. Toreyin et al. [11] kept on tracking the history of red channel for each pixel which is part of fire contour in a relative short time, and took them as the input of wavelet method. Apart from the changing of fire region's area and flicker, Hu et al. [21] employed the changing of fire region's roundness which describes complexity of the shape, to help filter the regions with regular shape.

Different with the other kinds of fire surveillances, forest fire monitoring has its own properties. The cameras are usually installed on the top of mountains, and they are not very stable because of wind blow. View range of the cameras is relatively wide, can be 3-5 km generally or even about 8 km. Focal length of the cameras is changeable, and the size of objects in recorded images is not constant. Most of the published papers worked on detection and analysis of sole fire region, but in forest fire, there may be more than one flame regions in the monitored area. All of them have caused a great deal of trouble for vision based fire detection, therefore it is necessary to specially study the case of forest fire recognition.

Our proposed forest fire detection algorithm considers static and dynamic features subsequently. The rest of our paper is organized as: color based segmentation including 3D color model with GMM and colors labeling with new flame pattern definition are provided in section 2, computation of static features and SVM based classification are described in section 3, shape based matching of multiple regions among continuous video images is given in section 4, dynamic features computation and based final determination with SVM are provided in section 5, experimental results on images and videos are described in section 6, and the conclusion is given in section 7.

2. Color Based Fire Segmentation

2.1. 3D Color Model

For segmentation of possible flame regions, color values of each pixel in an image are checked with a pre-determined color distribution, which represents the range of possible fire colors in a color model such as RGB space. As shown in Fig. 1, there are 530,000 flame pixels segmented manually from the

fire regions of 23 sample images. Of course, the threshold values along R, G and B axis can be used to define a rough space for fire color. To build a more precise color model, 3D shape of the point cloud is represented by Gaussian mixture model (GMM), and the pixel whose color within the range of the GMM distribution model can be taken as a candidate fire pixel.

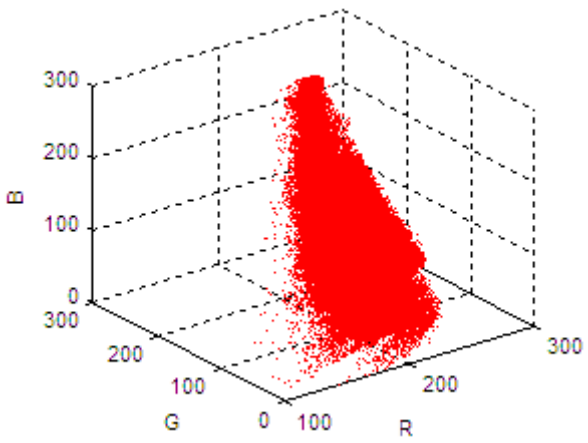


Fig. 1. 3D point cloud of sample fire pixels

First, we use expectation maximization (EM) algorithm to train the GMM parameters: the weight values, the center, and the covariance matrix. Then, whether one pixel belongs to flame region of the image under processing can be decided by calculation of its probability with the following formula:

$$g(x;\mu,\Sigma)=\frac{1}{\sqrt{(2\pi)^{d}|\Sigma|}}\exp\left[-\frac{1}{2}(x-\mu)^{T}\Sigma^{-1}(x-\mu)\right] \quad (1)$$

$$p(x)=\sum_{i}\alpha_{i}*g(x;\mu_{i},\Sigma_{i}) \quad (2)$$

The weighting value, center and covariance matrix of each Gaussian model are α, μ and $\sum$ respectively. The probability p(x) illustrates how close of point x to the fire region. The number of Gaussian models in GMM can be manually assigned or automatically computed. We tried automatic approach in experiments and the calculated best number of Gaussian models is 8. Based on the trained 3D color distribution model, pixels of one image are checked one by one, and then the possible flame regions within the image can be segmented automatically.

2.2. New Definition for Forest Flame Pattern

Ref. [20] studied color, geometry and motion of fire for recognition, and modeled the fire region in a single image as: (1) it stands in high contrast to its surroundings; (2) it exhibits a structure of nested rings of colors, changing from white at the core to yellow, orange and red in the periphery. This description gives a standard to detect fire. However, with respect to the case of forest flame, many fire regions do not have the structure of obvious nested rings, as shown in Fig. 2, and mostly because that they fail to burn fully.

Fig. 2. Fire regions with non-significant nested structure

Therefore, for forest fire, we present a new definition to describe the flame pattern more properly: (1) the periphery of fire region is in orange or red color; (2) only if the fire burns fully, there are one or more white-yellow color cores.

2.3. Labeling Three Types of Colors

Based on our new definition for forest fire pattern, there may be three types of colors in the segmented fire region: white-yellow, orange, red. Thus the pixels in fire can be labeled with three corresponding marks. As pixels with white-yellow color belong to the high bright flame regions, V value of the HSV color space is employed to help label such pixels. Since flames are often covered with smokes in forest, its bright value will be decreased with different degrees. In this case, we can not use a fixed V value as the threshold. Therefore, an algorithm is proposed to self-adaptively calculate the threshold value of V for fire images, and the algorithm is described as follows.

Two threshold values, V_{low} and V_{high}, are defined for V based segmentation. The lower threshold V_{low} is a constant value from our experimental experience, while the higher threshold V_{high} is a value computed automatically. Related with V_{low} and V_{high}, there are two subset of one image: L, set of the pixels whose V value is no less than V_{low}; H, set of the pixels whose V value is no less than V_{high}.

$$L = \{p_i \mid p_i \in [V_{low}, 255]\}$$
$$H = \{p_i \mid p_i \in [V_{high}, 255]\} \quad (3)$$

Obviously, H is a subset of L. If L is taken as the possible fire region, H can be used for further determination of fire cores. For the number of pixels, the percentage of H in L can be controlled by a parameter α

$$\frac{Num(H)}{Num(L)} < \alpha \tag{4}$$

The smaller the value of α, the less the number of pixels in H, together with the higher threshold value of V_{high}. The procedure to compute V_{high} is:

Step 1: Define the distribution function F(x) for subset L as

$$F(v(p_i)) = P(t \le v(p_i)) = \sum_{t=V_{low}}^{v(p_i)} f(t), p_i \in L \tag{5}$$

Step 2: Find the subset H based on F(x) as

$$H = \{p_i \mid F(v(p_i)) > 1-\alpha\} \tag{6}$$

Step 3: Calculate the higher threshold value V_{high} as

$$V_{high} = \min(v(p_i) \mid p_i \in H) \tag{7}$$

Once we get the higher threshold value $V_{high,}$ we label the pixels as white-yellow if their V value exceeds $V_{high.}$ For the left pixels of the candidate fire region, we use the experiential value of H and S ranges to decide whether they belong to orange or red.

3. Static Features and SVM Classifier

After color based segmentation, the possible flame regions are obtained in one single image. In our method, they are not directly used as fire areas, but are further checked to filter out the false candidates based on some static features with trained support vector machine (SVM). The static features include color distribution, texture parameter and shape roundness.

3.1. Static Features from Single Frame

(1) Color distribution (5 features)

During color segmentation with our new flame pattern, regions with white-yellow, orange and red color are labeled. We calculate the ratio of the pixels' number in each labeled color to the pixels' number in the entire candidate fire region. The ratio of white-yellow pixels (Eq. 8), the ratio of red pixels (Eq. 9) and the ratio of orange pixels (Eq. 10) are all static features of forest fire.

$$\text{Ratio}_{WY} = \frac{num(white - yellow)}{num(entire - region)} \tag{8}$$

$$\text{Ratio}_{R} = \frac{num(red)}{num(entire - region)} \tag{9}$$

$$\text{Ratio}_{O} = \frac{num(orange)}{num(entire - region)} \tag{10}$$

With respect to each candidate fire region, their color histograms in different color channels are computed respectively. Suppose I is one gray value in one color channel, Ng is the number of gray levels, P(I) is the ratio of the number of pixels with I value to the number of pixels in the candidate region, the expectation of Eq. (11) and the variance of Eq. (12) are computed as static features, e.g. two features for H color channel.

$$e = E[I] = \sum_{I=0}^{Ng-1} I * P(I) \tag{11}$$

$$\text{var} = E[(I - E[I])^2] = \sum_{I=0}^{Ng-1} (I - e)^2 P(I) \tag{12}$$

(2) Texture parameter (5 features)

Forest fire also has texture features [22-24], thus we can extract the texture parameters from each candidate region and then consider them in decision. Since H value represents the color information in HSV color space, the co-occurrence matrix of the region's H channel is employed to describe the texture feature. From experiments, it can be found that only the co-occurrence matrixes with zero degree have evident differences between fire and non-fire regions, thus parameters of zero degree co-occurrence matrix is used. Among parameters of the N*N co-occurrence matrix, the angular second moment of Eq. (13), the entropy of Eq. (14), the mean of Eq. (15), the contrast of Eq. (16) and the inverse difference moment of Eq. (17) are chosen as static features.

$$\text{Angular Second Moment} \quad \text{ASM} = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} p(i, j)^2 \tag{13}$$

$$\text{Entropy} \quad \text{E} = -\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} p(i, j) * \log_2 p(i, j) \tag{14}$$

$$\text{Mean} \quad \text{M} = \sum_{i=0}^{N-1} \text{i} * p(i) \tag{15}$$

$$\text{Contrast} \quad \text{C} = \sum_{n=0}^{N-1} n^2 \left\{ \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} p(i, j), |\text{i - j}| = n \right\} \tag{16}$$

Inverse Difference Moment $\mathrm{IDM} = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} p(i,j)/[1+(i-j)^2]$ (17)

(3) Shape roundness (1 feature)

Given a segmented candidate fire region, we retrieve its boundary using the classical Laplacian operator of $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$, then it is convenient to compute the 8-connected boundary chain code [25,26] for the region, as illustrated in Fig. 3. From the chain code, it is easy to calculate area S of the region and perimeter L of the boundary. Accordingly, we compute the shape roundness as L^2/S, which can describe the complexity of shape, i.e. more complex shape has larger soundness value. Shape roundness helps to get rid of the candidate regions less complex than fire, e.g. regular red car in image.

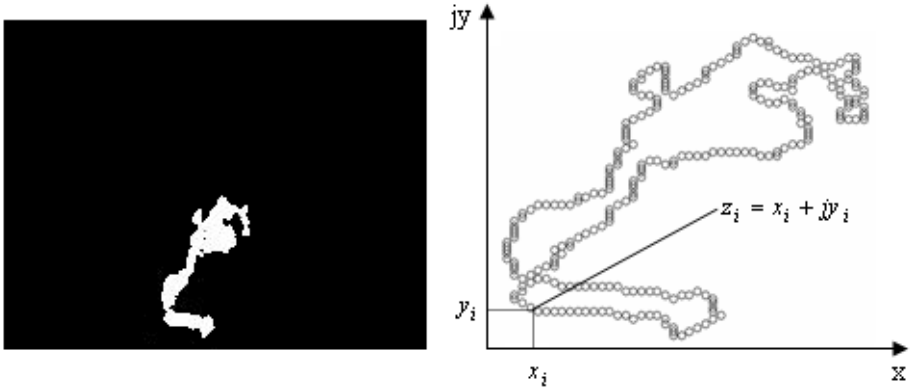


Fig. 3. Candidate region with its boundary chain code

3.2. SVM Classifier with Static Features

Support vector machine (SVM) is a set of related supervised learning methods that analyze data and recognize patterns, thus it is employed in our method for features based classification. The open source package (LIBSVM) is used to construct a two-class SVM classifier. To train the SVM, the above 11 static features are computed and collected from sample images with real fire or fire like objects. With the help of these features and radial basis function kernel, we can obtain the main parameters C=299.25 and γ=0.2295 for SVM. Therefore, the segmented candidate fire regions are further checked by the trained SVM classifier, and the false regions can be deleted. Of course, static features can help filter the candidate regions segmented from one

single image, but are not enough to describe the forest fire changing in video sequences.

4. Shape Based Flames Matching

Before computing dynamic characteristics of varying fire, the corresponding candidate fire regions should be found among consecutive video frames, which is a problem of pattern matching. Although the camera may mildly wobble and the candidate flame regions may randomly flicker, locations and shapes of the corresponding candidate regions among consecutive video frames do not change seriously. Therefore, two parameters of overlapping degree and varying degree are defined to evaluate the matching of two regions in our approach. Suppose R1 and R1' (or R2 and R2') are the corresponding regions (there may be multiple matching pairs) in two neighbor frames, as illustrated in Fig. 4.

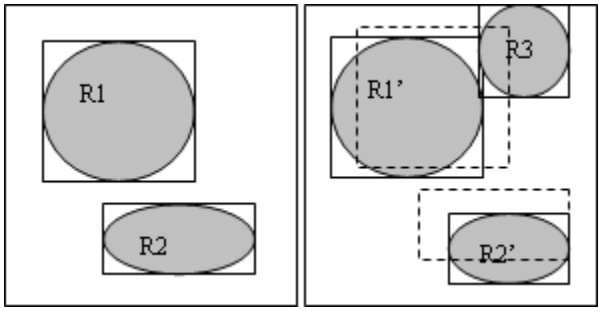


Fig. 4. Region matching of consecutive frames

The overlapping degree of two regions is

$$\frac{S(R_1 \cap R_1')}{\max(S(R_1), S(R_1'))} > \alpha, \quad 0 \le \alpha \le 1 \tag{18}$$

where $S(R_1 \cap R_1')$ represents the overlapping area of region R_1 and region R_1', $\max(S(R_1), S(R_1'))$ represents the larger area of R_1 and R_1', and α is an experiential value, i.e. the two corresponding regions are more like the same flame with the larger value of overlapping degree.

The varying degree of two regions is

$$\frac{\left|S(R_1) - S(R_1')\right|}{\min(S(R_1), S(R_1'))} < \beta, \quad 0 \le \beta \le 1 \tag{19}$$

where $S(R_1)$ represents the area of region R1, $S(R_1')$ represents the area of region R_1', $\min(S(R_1), S(R_1'))$ represents the smaller area of R_1 and R_1', and β is an experiential value, i.e. the two corresponding regions are more like the same flame with the less value of varying degree.

Fig. 5. Matching results of five fires from continuous frames

The proposed definitions for flames matching are tested by some collected video clips. As shown in Fig. 5, the first row displays the source images, the second row refers to results from color segmentation, the third row illustrates the matching results. In the matched results, fire regions with the same color mean that they are the corresponding regions of the same flame.

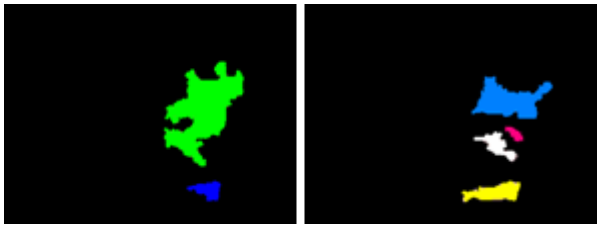


Fig. 6. Detection of the breaking behavior

Behaviors of forest fire are very complex with time, e.g. one fire region can slowly break into several small parts, or the small fires can burn into one region. Our matching algorithm has the ability to detect such changes. As illustrated in Fig. 6, one flame divides into several small regions, and the small parts cannot match with the whole fire. In this case, the number of candidate fire regions increases and the matching operation is performed on the new set of regions.

5. SVM Determination with Dynamic Features

Based on the matched results, dynamic features of the candidate fire regions from continuous video frames can be extracted, and used to further identify forest fire from the other fire like objects. In our method, the dynamic features include the variations of color distribution, texture, roundness, area, contour and the flickering frequency.

5.1. Dynamic Features from Matched Regions

(1) Variation of color distribution (5 features)

It is defined as the variation of color distributions (from Eq. 8 to Eq. 12 in section 3.1) of one candidate region among a sequence of video frames.

(2) Variation of texture (5 features)

It is defined as the variation of texture parameters (from Eq. 13 to Eq. 17 in section 3.1) of one candidate region among a sequence of video frames.

(3) Variation of roundness (1 feature)

It is defined as the variation of shape roundness (in section 3.1) of one candidate region among a sequence of video frames.

(4) Variation of area (1 feature)

It is defined as the variation of area of one candidate region among a sequence of video frames. Area is represented by the number of fire pixels in one region, and the area of forest fire is continuously changing since fire is an instable and developing procedure.

(5) Variation of contour (1 feature)

It is defined as the variation of contour of one candidate region among a sequence of video frames. Since the shape of fire region is changeable owing to air flowing, we can calculate the contour fluctuation to measure the disorder. Assume there are N points on the boundary, and they are expressed in the complex form $\{z_i \mid z_i = x_i + jy_i\}$, where (x_i, y_i) are the coordinates of the ith point on the boundary traversed clockwise, as shown in Fig. 3. Coefficients of the discrete Fourier transform (DFT) [26,27] of z_i are then calculated as

$$F_w = \frac{1}{N}\sum_{i=1}^{N} z_i \exp(-j\frac{2\pi}{N}iw) \tag{20}$$

where F_0 represents the centre of gravity of the transformed 1D boundary, which does not carry shape information, so we neglect it to achieve the

translation invariance. Experiments show that only a few dozens of the Fourier coefficients are really needed to describe the contour, thus the front 32 ones $D = (\left\| F_1^{'} \right\|_2, \left\| F_2^{'} \right\|_2, ... \left\| F_{32}^{'} \right\|_2)$ are used, and the difference of two consecutive Fourier descriptors corresponding to two neighbor frames is defined as

$$D_i = \sum_{w=1}^{32} \left\| \left\| F_w^{i'} \right\|_2 - \left\| F_w^{i-1'} \right\|_2 \right| \tag{21}$$

If D_i is greater than Td and lasts for a time period longer than Tm, where Td and Tm are statistical threshold values from experiments, it means that there is a drastic change in shape and the region is probably a fire.

(6) Flickering frequency (1 feature)

Flickering frequency is another important clue for forest fire since the flames flicker with a characteristic frequency around 10 Hz. We compute the variance of every two consecutive Fourier descriptors in a relatively short time and then analyze the sequence of variances with temporal wavelet. The video capturing rate should be high enough to capture flame flickering, i.e. at least 20 Hz to deal with the 10 Hz fire flickering. In our experiment, the digital camera can capture 30 frames per second, so it works.

As shown in Fig. 7, $x_n[k, l]$ represents the variance of Fourier descriptors between the nth and the (n+1)th frame, and each $x_n[k, l]$ in a relatively short time is assigned to a two-stage filter bank. The two-channel decomposition filter is constituted of high-pass filter (HPF, {-0.25, 0.5, -0.25}) and low-pass filter (LPF, {0.25, 0.5, 0.25}). If there is high frequency variation, high-band sub-signals d_n and e_n should be non-zero value. On the contrary, if the nth frame stay stationary compared with the consecutive frame, these two sub-signals should be equal to zero or very close to zero due to the high-pass filters. Thus the number of zero crossings of the sub-band signals d_n and e_n in one period is used as fire flickering frequency.

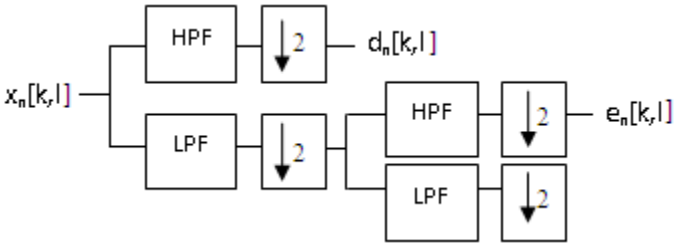


Fig. 7. A two-stage filter with HPF and LPF

5.2. SVM Classifier with Dynamic Features

Since the flickering frequency is a constant value of about 10 Hz, it can be directly used as one dynamic feature of forest fire. For color distribution, texture, roundness, area and contour, variations of them (13 features) from n consecutive images are computed and then taken as dynamic features.

To make sure that fire detection performs in real time and gives alarms without delay, n should be a relative small number. Based on the fact that the flames flicker around 10 Hz and the recorded videos have 30 frames per second, n is assigned with the value of 20. That is, dynamic features are computed for the forest fire from every 20 consecutive video frames. Therefore, an n*m matrix is constructed for fire features of a video clip, where n=20 while m=13 is the number of aforementioned dynamic features. Suppose $X(i, j)$ is one element of the matrix corresponding to the ith video frame and the jth fire feature, dynamic features based on the matrix are defined as the average

$$E(j) = \frac{1}{n}\sum_{i=1}^{n} X(i, j) \tag{22}$$

and the mean square deviation

$$S(j) = \sqrt{\frac{1}{n}\sum_{i-1}^{n}(X(i, j) - E(j))^2} \tag{23}$$

Therefore, for any video clip, there are 2*13=26 varying dynamic features, i.e. the average and the mean square deviation of color distribution, texture, roundness, area and contour. Together with the flickering frequency, the 27 dynamic features are used as input of SVM classifier, and the trained SVM is applied for the final decision.

6. Experimental Results

We developed our algorithm with C++ and Open CV under VC .NET in Windows XP, and tested it on a lot of video clips with real fires or fire like objects. As our algorithm includes color based segmentation and SVM based static and dynamic classifications, not only the final recognition, but also the intermediate results are displayed and analyzed.

6.1. Results of Color Based Segmentation

As shown in Fig. 8, video frames (the 1st, 2nd and 3rd columns) and still images (the 4th column) in the 1st row are collected and used as the experimental data. For comparison, the color based segmentation method of

Ref. [7] is tested first. Results from the first group of inequations in [7] with relative loose limits are shown in the 2nd row, and it can be found that many other regions remain as they have the similar color as fire. Results from the second group of inequations in [7] with relative strict limits are shown in the 3rd row, and it can be found that only a few real fire regions are segmented.

Fig. 8. Segmentation results from [7]

Then another color based segmentation method of Ref. [8] is tested based on its three deduced decision rules. In their method, parameters RT and ST of the decision rules must be set manually and they exert crucial influence on fire segmentation, especially for parameter ST. As shown in Fig. 9, the 1st row displays the segmented results from RT=170 and ST=0.3, while the 2nd row displays the segmented results from RT=170 and ST=0.9.

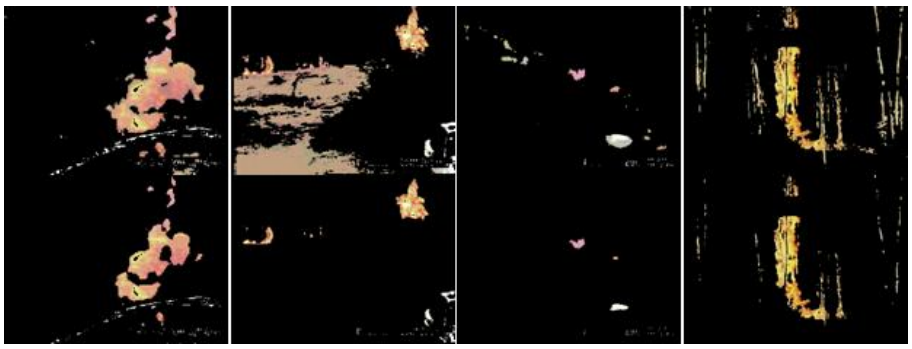


Fig. 9. Segmentation results from [8]

The same images are used to test our method, as shown in Fig. 10. From the experimental results it can be found that our algorithm has the ability to segment fire regions more precisely from the monocular images, and the regions with white-yellow, orange, red colors are also labeled respectively illustrated with different gray value, i.e. the segmented regions from our approach has more information. Of course, the segemented results are only possible fire regions, and they need further determination.

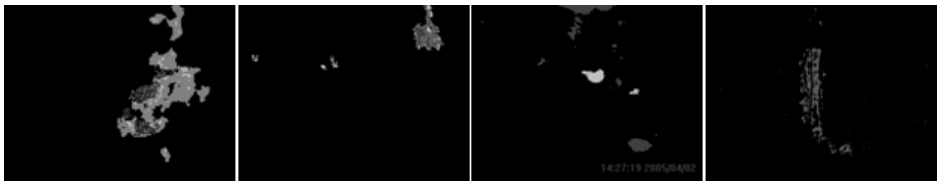


Fig. 10. Segmentation results from our method

Fig. 11. Segmentation results of fire colored objects

Our method is also tested on some images with fire colored non-fire objects such as bright light, flying red flag and moving red car. From Fig. 11, it can be found that the objects with fire like color can be taken as possible fire regions if using only color based segmentation, which also proves the necessity to take the other fire features into consideration except for color.

6.2. Determination with Static and Dynamic Features

The static features defined in our method are used for further determination with the help of SVM classifier. Results of Fig. 12 illustrate that the static features can help remove a lot of fire like regions since there are differences between their color distributions, texture parameters or shape roundness and those of real fire. But for the flying red flag, even its static features are very similar with forest fire, thus the flag still remains after static SVM. Then the dynamic SVM classification is performed on continuous video frames with the dynamic features including variations of color distribution, texture, roundness,

area, contour and the flickering frequency. After dynamic SVM, the flying red flag is recognized and filtered.

Fig. 12. SVM determination of fire like regions

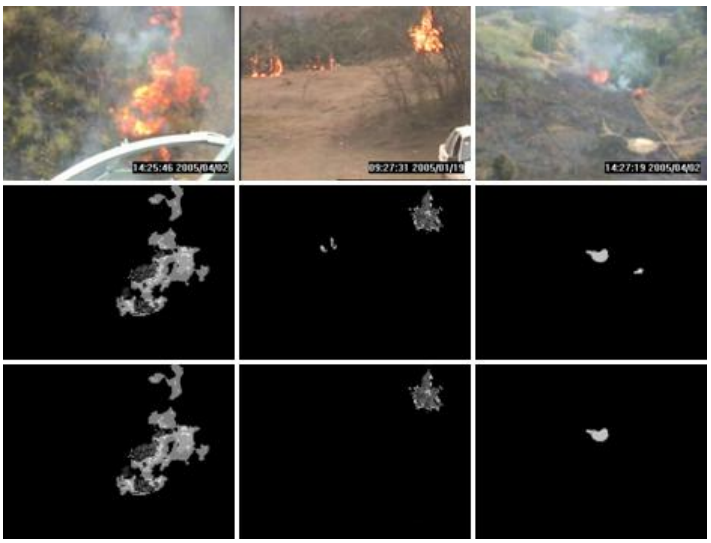


Fig. 13. SVM determination of real fire regions

With the same parameters, SVM classifications using static and dynamic features are consequently performed on the segmented results of Fig. 10 with real forest fire. Since the image of the 4th column in Fig. 10 is a single image and dynamic SVM cannot be applied, only the classification results of the frames from video clips (the 1st, 2nd and 3rd columns) are shown in Fig. 13. It can be found that the main fire regions are detected successfully, but the small fire regions may be filtered. The reason is that there are fire features in the small regions represented by only a few pixels, but the features are relative weak compared with the main regions.

6.3. Flame Detection from Video Clips

Ref. [10] also presented a complete procedure for video based automatic fire detection, so the method is implemented and compared with our approach. The experiments are executed on dozens of video clips, and 8 of them are illustrated in Fig. 14. Data of the first row are videos from F1 to F4 with forest fire, while data of the second row are videos from N1 to N4 with fire similar objects. For video clips with fire, F1 is early fire, F2 is fully burning fire, F3 and F4 are fire covered with thin smoke. For video clips without fire, N1 is a moving red car, N2 is a red house captured with shaking camera, N3 is a bright driving light, and N4 is a red flag flying in the wind. That is, the data can test the detection performance under different situations.

Fig. 14. Fire detection from video clips

Table 1. Performance comparison between Ref. [10] and our method

Videos	Total frames	Fire frames	Alarm frames of Ref. [10]	Alarm frames of our method	Alarm ratio (%) of Ref. [10]	Alarm ratio (%) of our method
F1	154	154	33	85	21.43	55.19
F2	1602	1602	717	1489	44.76	92.95
F3	192	192	115	183	59.90	95.31
F4	450	450	98	187	21.78	41.56
N1	262	0	114	6	43.51	2.29
N2	611	0	367	31	60.07	5.07
N3	192	0	191	1	99.48	0.52
N4	2072	0	1181	67	57.00	3.23

The performance comparisons between Ref. [10] and our approach are shown in Table 1. The 1st column lists the names of video chips in our experiment; the 2nd column and the 3rd column list the total frames and real fire frames respectively; the rest columns list the alarm frames and alarm ratios from two compared methods. For video clips with real fire, our method

gives better detection accuracy with more alarm ratios. For video clips with fire like objects, our method provides higher recognition precision with less alarm ratios. Of course, our approach has lower accuracy for fire with small regions (F1), and the performance is even worse for small fires covered by smoke (F4). Our algorithm runs on a PC with a CPU C1.7G and 512M DDR RAM, and has a speed of 33.02 fps. Therefore, except for the acceptable accuracy, our method can perform in real time.

7. Conclusion

In this paper, a new SVM based approach is proposed for forest fire detection with both static and dynamic features. Compared with the published related works, our novel method has the following advantages:

(1) In color based segmentation, after GMM construction from sample pixels and segmentation of candidate fire regions, we define the specific forest flame pattern and then label three types of colors including white-yellow, orange and red. The labeling introduces a novel feature of forest fire, i.e. color distribution, which is very helpful for further classification.

(2) For the segmented results from single frame, SVM trained on 11 static features is applied to filter out the false regions, and only the remained regions continue with the following steps. In this way, computational expense is saved obviously.

(3) Not only the sole target, but multiple candidate fire regions are tracked by shape based matching among the consecutive frames. With our defined overlapping degree and varying degree, the matching algorithm can also detect complex fire behaviors, e.g. one fire region slowly breaks into several small parts, or the small fires burn into one region.

(4) To compute the fire flickering frequency based on region contour, the temporal wavelet is used to analyze Fourier descriptors representing the variation of flame contour in a short period. Our approach avoids explicitly setting the threshold value in the existing FFT methods, while detects forest fire more accurately than the methods using wavelet transformation only.

(5) A total of 27 dynamic features are considered for SVM based final classification, and the features are computed from every 20 consecutive video frames. Therefore, except for accuracy, the detection algorithm can perform and give alarms in real time.

Our work has been tested with a lot of real video clips and the experimental results have proved its efficiency. However, for fire with small regions or fire regions covered with smoke, there are relative poor static and dynamic fire features, and thus the detection accuracy is still low. In the future, we will try the other ways for such problem, e.g. segmenting smoke first, and considering both fire and smoke together.

Acknowledgments. This work was supported by Hubei Provincial Natural Science Foundation of China, National Basic Research Program of China (973 Program, No. 2011CB707904), Fundamental Research Funds for the Central Universities, Research Foundation (No. AISTC2008_16) from the State Key Laboratory of Aerospace Information Security and Trusted Computing of Ministry of Education, and 985 Project of Cognitive and Neural Information Science, Wuhan University (No. 904273258).

References

1. Lai C.L., Yang J.C., Chen Y.H.: A Real Time Video Processing Based Surveillance System for Early Fire and Flood Detection. Instrumentation and Measurement Technology Conference, pp. 1-6. (2007)
2. Toreyin B.U., Cetin A.E.: Online Detection of Fire in Video. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-5. (2007)
3. Yuan F.N.: A fast accumulative motion orientation model based on integral image for video smoke detection. Pattern Recognition Letters, 29, pp. 925-932. (2008)
4. Han D., Lee B.: Flame and smoke detection method for early real-time detection of a tunnel fire. Fire Safety Journal, 44, pp. 951-961. (2009)
5. Gunay O., Tasdemir K., Toreyin B.U., Cetin A.E.: Video based wildfire detection at night. Fire Safety Journal, 44, pp. 860-868. (2009)
6. Yu C., Fang J., Wang J., Zhang Y.: Video Fire Smoke Detection Using Motion and Color Features. Fire Technology, 46(3), pp. 651-663. (2009)
7. Celik T., Demirel H., Ozkaramanli H., Uyguroglu M.: Fire Detection in Video Sequences using Statistical Color Model. IEEE International Conference on Acoustics, Speech and Signal Processing, pp. II213-216. (2006)
8. Chen T.H., Wu P.H., Chiou Y.C.: An Early Fire-Detection Method Based on Image Processing. IEEE International Conference on Image Processing (ICIP), pp. 1707-1710. (2004)
9. Celik T., Demirel H.: Fire Detection in Video Sequences using a Generic Color Model. Fire Safety Journal, 44(2), pp. 147-158. (2009)
10. Phillips Ⅲ W., Shah M., Lobo N.V.: Flame recognition in video. Pattern Recognition Letters, 23, pp. 319-327. (2002)
11. Toreyin B.U., Dedeoglu Y., Gudukbay U., Cetin A.E.: Computer vision based method for real-time fire and flame detection. Pattern Recognition Letters, 27, pp. 49-58. (2006)
12. Krstinic D., Stipanicev D., Jakovcevic T.: Histogram-based smoke segmentation in forest fire detection system. Information Technology and Control, 38(3), pp. 237-244. (2009)
13. Martinez-de Dios J.R., Arrue B.C., Ollero A., Merino L., Gomez-Rodriguez F.: Computer vision techniques for forest fire perception. Image and Vision Computing, 26, pp. 550-562. (2008)
14. Ko B.C., Cheong K.H., Nam J.Y.: Fire detection based on vision sensor and support vector machines. Fire Safety Journal, 44, pp. 322-329. (2009)
15. Toreyin B.U., Dedeoglu Y., Cetin A.E.: Flame detection in video using hidden Markov models. IEEE International Conference on Image Processing (ICIP), pp. 1230-1233. (2005)
16. Celik T., Demirel H., Ozkaramanli H.: Automatic fire detection in video sequences. European Signal Processing Conference, pp. 1-5. (2006)

17. Chen T.H., Kao C.L., Chang S.M.: An Intelligent Real-Time Fire-Detection Method Based on Video Processing. IEEE 37th International Carnahan Conference on Security Technology, pp. 104-111. (2003)
18. Zhang J.H., Zhuang J., Du H.F.: A New Flame Detection Method Using Probability Model. International Conference on Computational Intelligence and Security, pp. 1614-1617. (2006)
19. Yuan F.N., Liao G.X., Zhang Y.M., Liu Y.: Feature Extraction for Computer Vision Based Fire Detection. Journal of University of Science and Technology of China, 36(1), pp. 39-43. (2006)
20. Liu C.B., Ahuja N.: Vision based fire detection. 17th International Conference on Pattern Recognition (ICPR), pp. 134-137. (2004)
21. Zhang D.Y., Hu A.K., Rao Y.J., Zhao J.M., Zhao J.H.: Forest Fire and Smoke Detection Based on Video Image Segmentation. SPIE Pattern Recognition and Computer Vision, pp. 67882H-1-67882H-7. (2007)
22. Ferrari R.J., Zhang H., Kube C.R.: Real-time detection of steam in video images. Pattern Recognition, 40, pp. 1148-1159. (2007)
23. Cremers D., Rousson M., Deriche R.: A Review of Statistical Approaches to Level Set Segmentation - Integrating Color, Texture, Motion and Shape. International Journal of Computer Vision, 72(2), pp. 195-215. (2007)
24. Liu X.W., Wang D.L.: Image and Texture Segmentation Using Local Spectral Histograms. IEEE Transactions on Image Processing, 15(10), pp. 3066-3077. (2006)
25. Arrebola F., Bandera A., Camacho P., Sandoval F.: Corner Detection by Local Histograms of Contour Chain Code. Electronics Letters, 33(21), pp. 1769-1771. (1997)
26. Zhang Z., Zhao J.H., Zhang D.Y., Qu C.Z., Ke Y.W., Cai B.: Contour Based Forest Fire Detection Using FFT and Wavelet. International Conference on Computer Science and Software Engineering, pp. 760-763. (2008)
27. Zhang D.Y., Han S.Z., Zhao J.H., Zhang Z., Qu C.Z., Ke Y.W., Chen X.: Image Based Forest Fire Detection Using Dynamic Characteristics With Artificial Neural Networks. International Joint Conference on Artificial Intelligence, pp. 290-293. (2009)

Jianhui Zhao received the B.Sc. degree in Computer Engineering from Wuhan University of Technology in 1997, the M.Sc. degree in Computer Science from Huazhong University of Science and Technology in 2000, and the Ph.D. degree in Computer Science from Nanyang Technological University in 2004. From 2003 to 2006, he worked as a Research Assistant/Associate in Hong Kong University of Science and Technology. Currently he is working as an Associate Professor in Computer School of Wuhan University. His research interests include digital image processing and computer graphics.

Zhong Zhang is a graduate student in Computer School of Wuhan University, and his research interests are image processing and pattern recognition.

Shizhong Han is a graduate student in Computer School of Wuhan University, and his research interests are pattern recognition and machine learning.

Chengzhang Qu is a PhD candidate in Computer School of Wuhan University, and his research interests are image processing and computer vision.

Zhiyong Yuan is working as an Associate Professor in Computer School of Wuhan University. His research interests include digital image processing and computer graphics.

Dengyi Zhang is working as a Professor in Computer School of Wuhan University. His research interests include embedded system design, image processing and pattern recognition.

Received: October 12, 2010; Accepted: January 17, 2011.

DOI: 10.2298/CSIS100411031F

The Trustworthiness Analyzing of Interacting Business Process Based on the Induction Information

Xianwen Fang[1,2], Changjun Jiang[1], Zhixiang Yin[2], and Xiaoqin Fan[1]

[1] Key Lab of Embedded System and Service Computing,
Ministry of Education, Tongji University,
Shanghai 201804, China
{fangxianwen, cjjiang,xqfan}@hotmail.com
[2] Department of Information and Computing Science,
Anhui University of Science and Technology,
Huainan 232001, China
{fangxianwen, zxyin62}@hotmail.com

Abstract. Under the open environments, it is very difficult to guarantee the trustworthiness of interacting business process using traditional software engineering methods, at the same time, for dealing with the influence of external factors, some proposed business process mining methods are only effective 1-bounded business process, and some behavior dependent relationships are ignore. A behavior trustworthiness analysis method of business process based on induction information is presented in the paper. Firstly, aimed to the internal factors, we analyze the consistent behavior relativity to guarantee the predictable function. Then, for the external factors, in order to analyze the behavior change of business process, we propose a process mining methods based on induction information. Finally, experiment simulation is given out, and compares our method with genetic process mining methods. Theoretical analysis and experimental results indicate that our method is better than the genetic process mining method.

Keywords: trustworthiness, consistent behavior relativity, business process, process mining.

1. Introduction

With the tremendous growth of information and communication technologies, it is advantageous to design and implement the complex inter-enterprise business processes. One of the major innovations is the concept of service-oriented architectures (SOA) which considers software systems as being made up with autonomous, dynamic, loosely coupled and service-based components. For interacting business processes, especially those that run

over the Internet, behavior trustworthiness plays a crucial role. The operating and development environment of business process turn from the traditional closed static environment and extend to an open, dynamic, ever-changing network environment [1]. In the network computing environment, the behavior of the entity is uncontrollable and uncertain, which posed a severe challenge for behavior trustworthiness of the running processes [2]. Therefore, the guarantee of the behavior trustworthiness of the business under the network environment has become the industry's research focus.

Analysts often see lack of trustworthy methods as a major impediment to the adoption of web services in building agile business process. Software behavior is a running operation series, which can describe the software some feature when interacting with others. By analyzing the definition of trustworthiness, that the behavior can be predictable is a key property. So, achieving behavior trustworthy goals often relies on an appropriate use of software behavior. However, the notion of behavior trustworthiness is often neglected in business process, which usually concentrates on modeling the process in a way that functional correctness and security can be shown, either manually or using formal proof tools like model checking and some static validation methods. In contrast to features that are crucial for functional correctness and security features, and so on, behavior trustworthiness is typically integrated in an application, which mainly assures the execution behavior and computing result of the business process to be consistent with the expected effects. The integration of trustworthy features into a business process is not well understood. Especially, the trustworthy features are difficult to describe, and to analyze quantitatively [3]. So, the problem need further investigation, and seek new methods to deal with the behavior trustworthiness.

The interacting business process may run into two kinds of problems, one problem is the interactive influence between the business processes in the building process; one component business process may impact the other, which leads to the function of composite business process to be improper. The other problem is that the composite business process may be influenced by the outer factors. Aimed to the internal interactive influence, the interaction behavior relativity of business process need be studied. For the outer factors, we use the business process mining methods to construct the actual behavior model based on the running logs. At present, some process mining methods [4-6] have been proposed, but the existing methods are deficient to solve the problem. In the paper, our contributions are given out the analyzing methods of the behavior trustworthiness about business process based on the induction information; the methods can deal with k-bounded business process, and take the indirect dependent relationships into account.

The rest of this paper is organized as follows. Section 2 introduces related work. In Section 3 discusses motivating examples and research framework. Section 4 details propose the analyzing of the composite model of business process based on Petri net. Section 5 presents Building the K-bounded process behavior model based on new process mining methods. Simulation

experiments are given out in the Section 6. Finally, Conclusions and acknowledgements are in Section7 and Section 8 respectively.

2. Related Works

The study of business process mainly takes advantage of some methods that described the interaction between different components of the system. Research on business process modeling has recently started to encode business-process diagrams into a formal model that can be given a suitable semantics, usually based on interacting state machines [7], Petri nets [8], or graph grammars [9]. Models that explicitly address the incorporation of security issues in the design process are typically extensions of a fragment of UML that can be given the desired semantics. They address more general notions of security than in the standards like multi-level security [10] or role-based access control [11]. A concrete counterexample is given in [12]. Being more precise, there is no guarantee that properties of these abstractions are also valid for the concrete implementation. In the literature [13], the paper presents a general methodology for integrating arbitrary trusted requirements in the development of business processes in a both elegant and rigorous way, and show how trusted relationships between different parties and their respective security goals can be reflected in a specification, which results in a realistic modeling of business processes in the presence of malicious adversaries. Special attention is given to the incorporation of cryptography in the development process with the main goal of achieving specifications that are sufficiently simple to be suited for formal verification, yet allow for a provably secure cryptographic implementation. [14] described a business process driven framework (called the BPD-ACS) for developing both the model and formulating the access decision rules. The model used is the Role Based Access Control (RBAC) model and the access decision rules are based on temporal business associations. In the literature [15], analysis of workflow dynamic changes based on Petri net is proposed, but did not take the behavior relativity into account.[16] proposed the logic-based approach about specification and verification of declarative open interaction of business process, in order to guarantee the trustworthiness and reliability of the developed model, as well as to ensure that the actual executions of the system effectively comply with it.[17-18] proposed a business process to be trustworthy if the behavior of all services is trustworthy, and gave out a verification mechanism through which the trustworthiness of a business process can be verified. Collaborative business processes often consist of services provided by multiple business entities which agree to join business collaboration. To enable trustworthy and secure consumption and provisioning of services across organizational boundaries, security requirements must be carefully defined so as to be coherent, consistent, and in compliance with designed business processes. However, managing security requirements in collaborative environments is error-prone, effort

inefficient, and hard to be verified. [19] discussed the architecture specific to this issue, as an add-on to a trustworthy service-oriented architecture, and proposed suitable formal notations and formal analysis in the construction of this automated facility.

All works mentioned above did not study the behavior trustworthiness of business process, only the function correctness or security, so it is necessary to study new methods to analyze the behavior trustworthiness of interactive business processes.

3. Motivating Examples and Research Framework

3.1. Motivating Examples

In order to satisfy the user' demand, it is necessary though compositing several business processes to realize the complex function demand, and satisfies certain constraints. Under open environment, some composite business process cannot be anticipated after the interacting, but the behavior predictability of the composite business process is a key to the trustworthiness study of business process. At present, some behavior conformance methods is proposed, the methods need to build the dynamic model based on the running log, and compare the dynamic model with the original theoretical model, obtain the behavior conformance by some metrics methods. In building the dynamic model, some process mining methods is proposed, such as @-algorithm [4], genetic mining methods[20,21]. These methods can solve effectively safe business process, but they are deficient to deal with K-bounded business process.

The ultimate objective of this paper is to study the behavior trustworthiness based on the process mining and induction information. The notion establishes whether the behavior of interacting business process is predictable and it can be employed to suitably address the following issues:

1) The process behavior models are built by some event dependent relationships and the k-bounded value. For example, the process behavior model may be not 1-bounded, so it is can not implement by @-algorithm. The region-based mining methods can implement the k-bounded model, but the methods obtain the k-bounded value by analyzing the transition system, which may lead to the error value and cost a mount of time-consuming.

2) Some indirect dependent relationships should take into account in the process of business process mining, but the region-based methods ignore some indirect dependent relationships.

3) For analyzing the behavior predictability under outer factors, the behavior conformance of two models is a key method, which is determined by structure fitness and behavior appropriateness, the effect should be distinguishable.

We can hence outline the main features that a suitable notion of behavior predictability of business process in open and changing environments should have, that is, the function correctness and behavior conformance of process behavior model. Indeed, it is reasonable that the correct process behavior model can be built by new process mining methods. The behavior conformance between the theoretical composite models with the building process behavior model is also key problem for analyzing the behavior predictability.

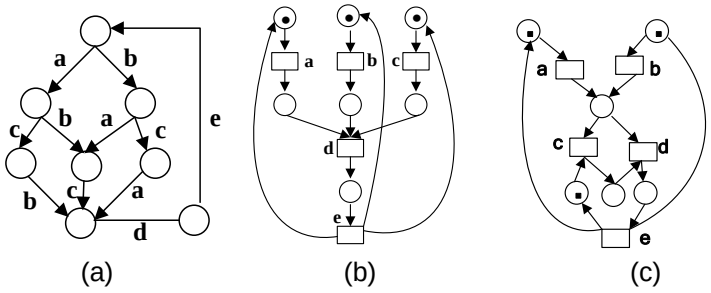


Fig.1. The TS and corresponding to process mining models.

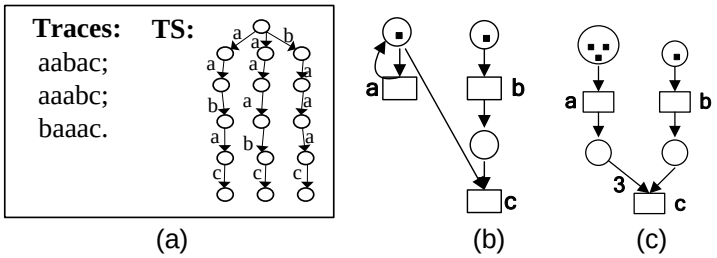


Fig.2. The log traces and corresponding to process mining modes

In Fig.1, Fig.1(a) is the transition system(TS) from the log traces of business process[22], which indicates the running state of business process, but it is transition running sequences without be handled, not the complete behavior relationships. So, it is necessary to mine the behavior relationships and to build the behavior model, in order to analyze the behavior conformance. Using the @-algorithm, we can obtain a soundness business process, the model is showed in the Fig.1 (b), because @-algorithm only can mine the 1-bounded model, the Fig.1 (b) is a 1-bounded model, and it can include all traces of the Fig.1 (a), but it generate more traces, for example the trace “cabde”, the Fig.1(a) can not generate it. In the Fig.1(c) is a 2-bounded business process model, its traces is the same as the Fig.1(a), so the Fig.1(c) is the better than the Fig1(b).

In Fig.2, Fig.2 (a) is the transition system from the log traces of business process,Fig.2(b) is 1-bounded process mining model by @-algorithm or others,Fig.2(c) is a 3-bounded process miming model. Obviously, the

language of Fig.2(b) is $(a^*\|b)c$, it is much greater than the traces in the Fig.2(a). The language of Fig.2(c) is $(a^3\|b)c$,it is greater than the traces in the Fig.2(a),but it is less than the language of Fig.2(b),and it is more accurate one, so the model in Fig.2(c) is better than the model in Fig.2(b).

Viewed from the Fig.1 and Fig.2, when we mine the process model, the bounded value is also key factor. It is necessary to analyze the bounded value before business process mining.

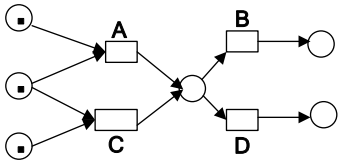


Fig.3. The process mining model of composite business process

In the Fig.3, for the log traces AB, CD of business process and the corresponding to transition systems TS{AB} and TS{CD}. When composing these business processes, it is easy to see that the minimal regions of each are compatible with the empty region of the other; hence the minimal regions of the composite transition system are simply the union of the two sets of minimal regions. The composite business process can be mined by region-based method proposed in the literature[22], the process mining model is showed in the Fig.3, we find that we can no longer just consider minimal regions, but also have to consider composite regions (in this case the region where A and C enter and B and D exit. So, the new region-based process mining methods proposed in the literature [22] can mined the k-bounded process model, but it has a drawback: incrementally considering only minimal regions does not always give a correct answer. This means that it can still construct the proper set of regions by considering the compatible ones, but we cannot simply always use the minimal regions of the components of the transition system.

3.2. Research Framework

Under the open and dynamic environments, owing to software evolution continually, it is very difficult to guarantee the software quality using traditional software engineering method; meanwhile, for dealing with external factors, there is a big limitation using the existing program verification and analyzing methods [12]. Because the existing formal verification methods mainly aimed to the program correctness in close environment, the composite software system may not be behavior trustworthy by some functional correctness software interacting with each other. With the increase of software scale and complex, and the process dynamic evolution, the traditional software testing technology is difficult to discover and locate

software vulnerability point. So it is difficult to guarantee that software is controllable, manageable, and preventable in open environment, also very difficult to realize the predictable behavior and effect. The software running behavior can reflect comprehensively the software behavior changing situation in the dynamic evolution and under the complex environment.

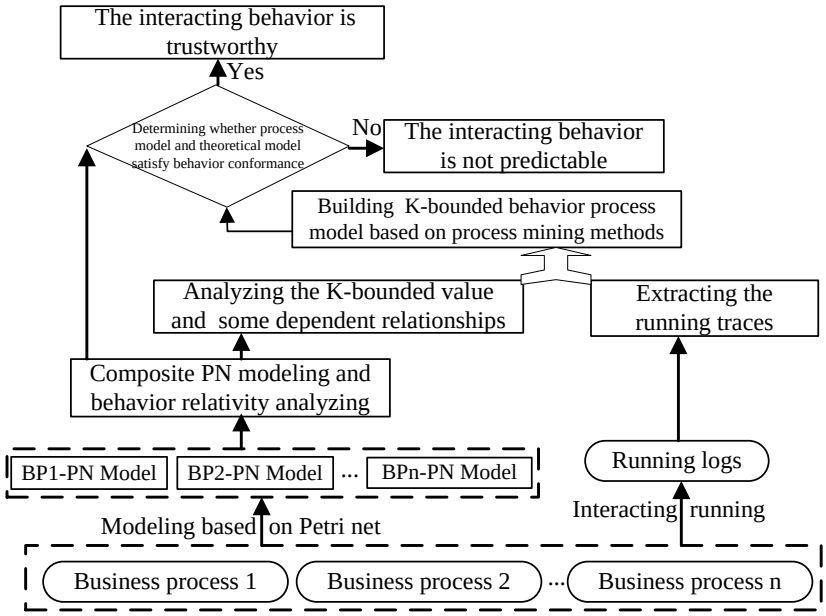


Fig.4. Research framework

By taking into account the requirements comprehensively outlined in this motivating section and the requirements of behavior trustworthiness, viewed from the software behavior, we propose a behavior-aware software trustworthiness research frame, showed in Fig.4. The main research idea is as follows: firstly, we model the software components based on Petri net, which can describe the system's dynamic characteristic concisely from the behavior angle, and has great advantage on describing some important activities and phenomena such as concurrence, conflict and synchronization, etc. Secondly, for analyzing the interacting influence of the software components, we take advantage of the analyzing methods of interaction behavior relativity to judge whether the interacting software are behavior consistent relativity. Finally, for dealing with the influence of external factors, we extract the dynamic behavioral model based on process mining, compare the dynamic behavior model with theoretical composite model, and judge whether the two models satisfy behavior congruence, if they are behavior congruence, then the component interacting process is not to be influenced by outside factors, such as virus, vicious program, and network environment and so on. The implementation process can be controllable, manageable,

and preventable, its behavior and result may be also predictable. If they are not complete behavior congruence, we analyze the behavior fitness and behavior appropriateness, and select an optimization composite behavior model.

4. The Analyzing of the Composite Model of Business Process Based on Petri Net

4.1. Basic Concept

Here, we only introduce several conceptions correlating with the paper close, other Petri Nets terms in the literature [23,24].

Definition 1[24].Let X be a finite alphabet, $Y \subseteq X$, let $\Gamma_{X\to Y} : X^* \to Y^*$,is called a projection from X to Y, if and only if $\forall \sigma \in X^*$,and $\Gamma_{X\to Y}(\sigma)$ is the residue string as all characters excluded from X-Y. The function $\Gamma_{X\to Y}$ is called the Projection from X to Y.

Definition 2[24].Let X be a finite alphabet, $Y \subseteq X$, let L_X , L_Y are the language of X and Y respectively.

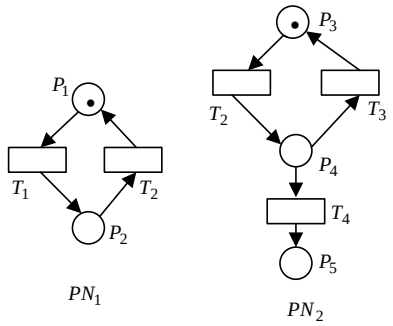


Fig.5. An example of consistent behavior relativity

If $\Gamma_{X\to Y} : (L_X) = \{\Gamma_{X\to Y}(\sigma) \in Y^* \mid \forall \sigma \in L_X\}$, then $\Gamma_{X\to Y} : (L_X)$ is called as the projection language of L_X from X to Y.

Definition 3[24]**.** Let $PN_i = (P_i, T_i; F_i, M_{0i})(i = 1,2)$ are two basic Petri nets, $PN = PN_1 O_T PN_2$. if $\Gamma_{T\to T_i}(L(PN)) = L(PN_i)$ (i=1,2), then called the behavior of PN_1 is consistent with PN_2 ,represented as $Be(PN_1) = Be(PN_2)$.

In Fig.5, $\Gamma_{T\to T_1}(L(PN)) = pref((t_1t_2)^*) = L(PN_1)$,

$\Gamma_{T\to T_2}(L(PN)) = pref((t_2t_3)^* + t_2t_4) = L(PN_2)$, so $Be(PN_1) = Be(PN_2)$.

4.2. The Function Predictability Analyzing of Composite Business Process Based on the Behavior Relativity of Petri Net

Component behavior relativity [24] mainly refers to one model behavior may be influence by others when component interacting, leading to some model behavior function occur to change, even some interactions are insignificance. The interaction behavior relativity has four kinds: consistent behavior relativity, interactive behavior relativity, controlled behavior relativity, exclusive behavior relativity. Consistent behavior relativity is one kind of good interactive behavior relationship, which indicates two interaction models accomplishing the function requirements and the behavior of themselves are not influenced. We can analyze whether the function and behavior of composite business process is influenced after interacting with each other, if they satisfy the consistent behavior relativity, the interaction process is good, which indicates the function and behavior are predictable. The decision algorithm of consistent behavior relativity is showed in algorithm 1.

Algorithm 1: The decision algorithm of consistent behavior relativity
Input: two Petri net models
Output: the result of consistent behavior relativity

Let $PN_i = (P_i, T_i; F_i, M_{0i})(i = 1,2)$ are two Petri nets, $PN = PN_1 O_T PN_2$, and $\Delta = T_1 \cap T_2$, $X_{ij_i}(j_i = 1,2,..., q_i'; i = 1,2)$ are all the minimum T-invariant of PN_i, $X_{ij_i}^{\Delta} = \Gamma_{T_i\to\Delta}(X_{ij_i})$, $j_i = 1,2,..., q_i, q_i \le q_i'; i = 1,2$ are the non-zero projection vector of the minimum T-invariant of PN_i, q_1 and q_2 are the number of PN_1 and PN_2 respectively.

(1) According to the definition of Petri net incidence matrix, computing the incidence matrix of PN_1 and PN_2.

(2) Computing the minimum T-invariant $X_{ij_i}(j_i = 1,2,..., q_i'; i = 1,2)$ of PN_1 and PN_2.

(3) Computing the projection vector $X_{ij_i}^{\Delta} = \Gamma_{T_i\to\Delta}(X_{ij_i})$,

$j_i = 1,2,..., q_i, q_i \le q_i'; i = 1,2$ of the minimum T-invariant of PN_1 and PN_2 on the share transition Δ.

(4) Computing the equation $X_{ij_i}^{\Delta} = \sum_{j_{3-i}=1}^{q_{3-i}} k_{3-ij_{3-i}} X_{3-ij_{3-i}}^{\Delta}$, $j_i \in \{1,2,..., q_i\}$,

$j_{3-i}=1,2,...,q_{3-i}$, $0\le k_{3-ij_{3-i}}\le 1$, $i=1\vee 2$. If the equation has non-zero solutions, the $k_{3-ij_{3-i}}$ is not all zero, which means that $X_{ij_i}^{\Delta}$ can be non-negative linear expressed by other some vectors, then $b(X_{ij_i}^{\Delta})=1$, otherwise, there only has zero solution, that is $k_{3-ij_{3-i}}=0$, which means that $X_{ij_i}^{\Delta}$ can not be non-negative linear expressed by other some vectors, then $b(X_{ij_i}^{\Delta})=0$.

(5) Determine whether the solution is all 1 , if it is "yes", then the projection of minimum T-invariant of PN_1 (PN_2) can be expressed by linear combination of the projection of minimum T-invariant of PN_2 (PN_1), so they satisfy the consistent behavior relativity, else, the result is not consistent behavior relativity.

For convenience, we develop a behavior relativity analyzer based on Petri net. The analyzer mainly including drawing Petri net, the matrix of Petri net, the language of Petri net, the composite Petri net and the analyzing of the interaction behavior relativity. The composite result of Fig.5 in analyzer is showed in the Fig.6.

4.3. Analyzing the K-bounded Value and Some Dependent Relationships

4.3.1. Behavior dependent relationship

For convenience, in this paper, the traces of event log are corresponding to the legal sequences of Petri net model, and taking the properties of Petri nets into account.

Definition 4[25](Basic order relationship) Let W is a event log, $PN=(P,T;F)$ is the building Petri net of business process, here, $W=L(PN)$, $a,b\in T$

（1） $a>_w b \quad iff\ \exists\sigma=t_1t_2t_3\cdots t_n\in W,\ \ i\in\{1,\cdots,n-1\}:t_i=a\wedge t_{i+1}=b;$

（2） $a\Delta_w b \quad iff\ \exists\sigma=t_1t_2t_3\cdots t_n\in W,\ \ i\in\{1,\cdots,n-2\}:t_i=t_{i+2}=a\wedge t_{i+1}=b;$

（3） $a\diamondsuit_w b \quad iff\ a\Delta_w b\ \wedge b\Delta_w a;$

（4） $a\rightarrow_w b \quad iff\ \ a>_w b\wedge(\neg(b>_w a)\vee a\diamondsuit_w b);$

（5） $a\ \#_w b \quad iff \neg(a>_w b)\wedge\neg(b>_w a);$

(6) $a\parallel_w b \quad iff\ a>_w b\wedge b>_w a\wedge\neg(a\diamondsuit b);$

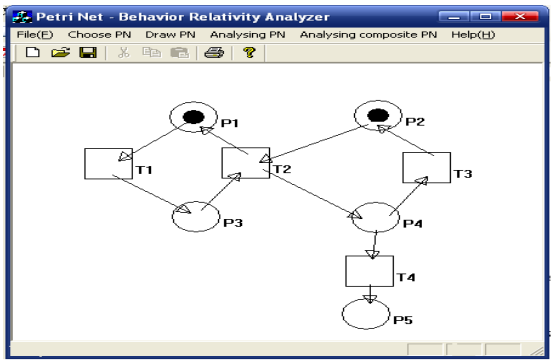


Fig.6. The composite result of Fig.5 in analyzer

As can be seen from the definition 4, $>_w$ and Δ_w are the basic order relationship, the former represents the two tasks occur one after another, the latter represents two tasks can generate a specific piece of the cycle track. These can be used to difference the relationships of two tasks. (3) (4) (5) (6) corresponding to cycle, sequence, select, and concurrency relations respectively.

Definition 5[25] (Direct dependent relationship) $PN=(P,T;F)$ is the building Petri net of business process, for arbitrary $a,b\in T$, if there exists direct dependent relationship between a and b iff

(1) $a^{\bullet}\cap{}^{\bullet}b\neq\Phi$;

(2)There exists reachability mark $s\in[PN,[i]>$ to make $(PN,s)[a>$ and $(PN,s-{}^{\bullet}a+a^{\bullet})[b>$

Definition 6[25] (Indirect dependent relationship) $PN=(P,T;F)$ is the building Petri net of business process, for arbitrary $a,b\in T$, if there exists indirect dependent relationship between a and b iff

（1） $a^{\bullet}\cap{}^{\bullet}b\neq\Phi$ and arbitrary $p\in a^{\bullet}\cap{}^{\bullet}b$, p is not the implicit place;

（2）There do not exist reachability mark $s\in[PN,[i]>$, to make $(PN,s)[a>$ and $(PN,s-{}^{\bullet}a+a^{\bullet})[b>$;

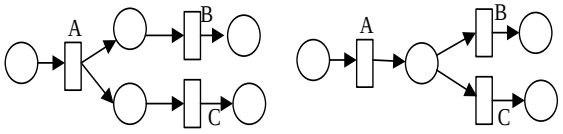
（3）There exists reachability mark $s\in[PN,[i]>$, to make $(PN,s)[a>$ and exists reachability mark $s'\in[PN,s-{}^{\bullet}a+a^{\bullet}>$,to make $(PN,s')[b>$.

4.3.2. The basic Petri net structure with the same sequence

The business process mining, is different from the existing workflow net mentioned in some literatures, because the business process may be not soundness, that is, the token number of the initial mark is also less than or equal to 1, but business process may be not 1 - bounded in the running

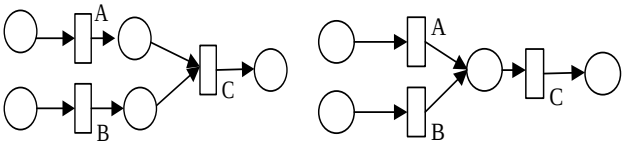
process, may remain some unused tokens after running. Business process mining can borrow some ideas from the workflow mining, but the structure of business process has its own specialty, the same sequence may correspond to different models. Now basic Petri net structures with the same sequence are given, it is convenient to learn new models in HMM-based process mining.

(1) If there exist AB and AC in two different traces, then it may be have two Petri net structures shown in Fig.7. Fig.7 (a) the corresponding structure has concurrency relationships between B and C, Fig.7(b) the corresponding structure has choice relationships between B and C.



(a) (b)

Fig.7. The corresponding Petri net structures of the sequences of AB and AC.



(a) (b)

Fig.8. The corresponding Petri net structures of the sequences of AC and BC.

(2) If there exist AC and BC in two different traces, then it may be have two Petri net structures shown in Fig.8. In Fig.8 (a) the corresponding structure has concurrency relationships between A and B, Fig.8(b) the corresponding structure has not concurrency relationships between A and B.

(3) If there exist ABD and ACD in two different traces, then it may be have two Petri net structures shown in Fig.9. In Fig.9(a) the corresponding structure has not concurrency relationships between B and C, Fig.9(b) the corresponding structure has concurrency relationships between B and C.

(4) If there exist ABD and ACD,ADB and ADC in the different traces, and satisfy the prerequisites, then it may be have Petri net structures shown in Fig.10, the corresponding structures have not concurrency relationships between B and C, have concurrency relationships between D and B or D and C.

(5) If there exist ACD and BCD in two different traces, then it may be have two Petri net structures shown in Fig.11. In Fig.11(a) the corresponding structure has concurrency relationships between B and C or A and C, Fig. 11(b) the corresponding structure has not concurrency relationships between A and B.

The above basic structures are some expansion structures to the direct dependent relationship of the basic order relationship. It can improve the learning process by using the corresponding basic Petri net structure with the same behavior sequence, but there exist some exceptional situation, a kind of indirect dependent relationship has not effect by using the basic structure. For example[25], the case sequences are {ACD, BCE}, if using a ordinary workflow mining method, corresponding model is shown in Fig. 12(a), but Fig. 12(a) model has the actual transition sequences are{ACD, BCE, ACE, BCD}, they are more than the existing sequences {ACD, BCE},so it is not a good mining result. But, if we take the indirect dependent relationship into account, and carry out process mining, the result is shown in Fig.12(b), which the actual transition sequences is the same with the case sequences are {ACD, BCE}. For the model with indirect dependent relationship, its corresponding basic Petri net structure is shown in Fig.13.

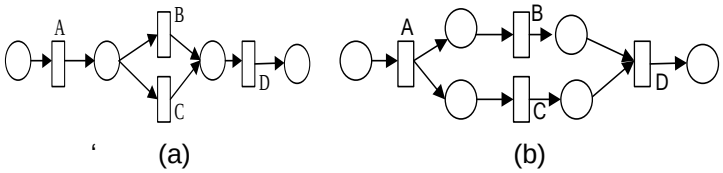


Fig.9. The corresponding two Petri net structures of the sequences of ABD and ACD.

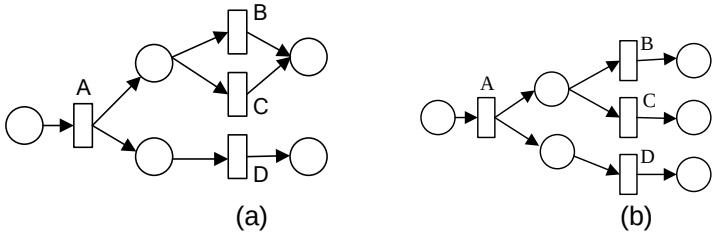


Fig.10. The corresponding Petri net structures of the sequences of ABD and ACD,ADB and ADC

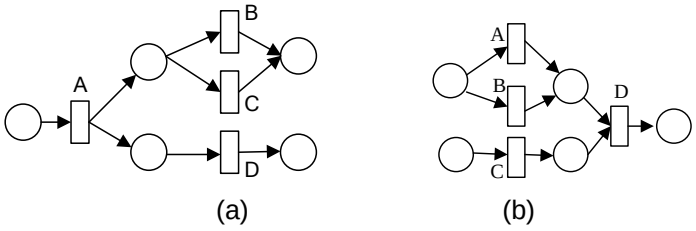


Fig.11. The corresponding Petri net structures of the sequences of ABD and BCD.

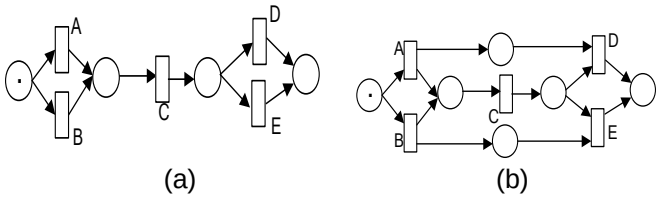


Fig.12. The corresponding Petri net model of the sequences of ACD and BCE.

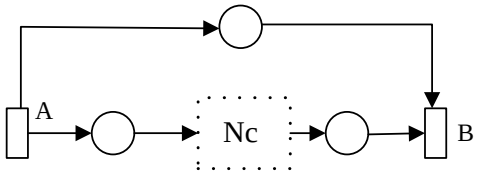


Fig.13. The corresponding Petri net structure which A and B are indirect dependent relationships

4.3.3. Analyzing the K-bounded value of the composite business process

In this paper, we need analyze the bounded value of the composite Petri net model, that is, determine the largest token number among all place of the composite Petri model, so we can adopt the reachability graph methods [23], firstly, build the corresponding reachability marked graph of Petri net, then computing the maximum value of all quantity of marking, the maximum value is the bounded value. The solving algorithm is as follows:

Algorithm 2: the bounded value of composite business process
Input: the Petri net model of business process
Output: the bounded value

(1) Aimed to the initial marking M_0, firing each enabled transition;

(2) Assume the current marking is M, firing the transition t of the current marking, and obtaining the marking M, adjusting every quantity of the marking, then continue to do;

(3) If M has appeared in the reachability graph, then draws a directed arc from M to the existing marking, if M has not appeared in the reachability graph t, but only one quantity of M marking is bigger than (smaller than) the corresponding quantity of the existing marking, then judges whether M increase (or decrease) by the times of running number. If “yes”, then abstract the M and the existing marking into a kind marking, the corresponding quantity is noted as “n”. Else, it is a new marking M',draws a directed arc from M to M', and adjusts the corresponding quantities;

(4) If the Petri net model has not the enabled transitions or can not generate new marking, then terminates the generating process;

(5) In all marking, solves the largest quantity among the reachability marking, and returns to this value, it is the k-bounded value. If the value is n, indicates that the Petri net model is unbounded.

5. Building the K-bounded Process Behavior Model Based on New Process Mining Methods

5.1. The Metrics of Behavior Conformance about the Petri Net Model

Generally, people use exemplary behavior to compare two models, the exemplary behavior can be obtained on the basis of real process executions, and the paper assumes this exemplary behavior to be recorded in an event log. To quantify differences between two models, we introduce structure fitness, behavior precision behavior recall and behavior redundancy metrics [20,26]. Structure fitness is used to measure the structure similarity of two models. Behavior precision is used to measure the cover degree which the second model's behavior includes the first model's behavior. Behavior recall is used to measure the cover degree which the first model's behavior includes the second model's behavior. Behavior redundancy is used to evaluate the occupancy of the high frequency traces. Using the mentioned-above metrics, we take some process mining tactics to obtain the better behavior model, which has high behavior precision and behavior recall based on the event log.

Definition 7 (The structure incident matrix of Petri net model) Let $PN=(P,T;F,M)$ is a Petri net, $P=(p_1,p_2,\cdots,p_m)$, $T=(t_1,t_2,\cdots,t_n)$, the structure incident matrix of Petri net model can be represented by a $n\times n$ matrix, the $A=(a_{ij})_{n\times n}$ (n is the transition number)

$$a_{ij}=\begin{cases}1 & \left|t^{\bullet}\right|=1, t_i^{\bullet}\cap^{\bullet}t_j\neq\Phi\\ 1/k & \left|t^{\bullet}\right|=k, t_i^{\bullet}\cap^{\bullet}t_j\neq\Phi\\ 0 & others\end{cases}$$

A is called as the structure incident matrix of Petri net model.

Definition 8 (Structure fitness) let structure incident matrix of PN_1 and PN_2 are $A_{n\times n}$, $B_{m\times m}$ respectively, suppose $n\geq m$,then the structure fitness of PN_1 and PN_2 is $1-\sqrt{\sum(a_{ij}-b_{ij}^{*})^2}\Big/n$,here, $(b_{ij}^{*})_{n\times n}=\begin{cases}b_{ij} & 1\leq i,j\leq m\\ 0 & others\end{cases}$ .If the structure fitness value is bigger, the structure fitness is better.

Definition 9[20] (Behavioral precision and recall). Let (N_1,M_1) and (N_2,M_2) be marked Petri nets and let $L \in B(T^*)$ be a multi-set over T.

$$precision((N_1,M_1),(N_2,M_2),L) =$$
$$(\sum_{\sigma\in L}\frac{L(\sigma)}{|\sigma|}(\sum_{i=0}^{|\sigma|-1}\frac{|enabled(N_1,M_1,hd(\sigma,i))\cap enabled(N_2,M_2),hd(\sigma,i)|}{|enabled(N_2,M_2,hd(\sigma,i)|}))/|L|$$
$$recall((N_1,M_1),(N_2,M_2),L) =$$
$$(\sum_{\sigma\in L}\frac{L(\sigma)}{|\sigma|}(\sum_{i=0}^{|\sigma|-1}\frac{|enabled(N_1,M_1,hd(\sigma,i))\cap enabled(N_2,M_2),hd(\sigma,i)|}{|enabled(N_1,M_1,hd(\sigma,i)|}))/|L|$$

Here, $hd(r,k) =< a_1,a_2,\cdots,a_k)$, i.e., the sequence of just the first k elements.

Definition 10 Assume the firing probability of all transition sequences is the same, the largest frequency of each sequence is the MAX, $\mathrm{MAX}=\max\{L(\sigma)|\sigma\in(N,M)\ \&\ \sigma\in L\}$, then the redundancy degree of Petri net based on the event log is

$$RED((N,M),L) = \left(\sum_{\sigma\in L}\frac{MAX-L(\sigma)}{|\sigma|}\left|\{i\in\{0,|\sigma|-1\}|\sigma(i+1)\notin enabled(N,M,hd(\sigma,i))\}\right|\right)\Big/|L|$$

Here, $hd(r,k) =< a_1,a_2,\cdots,a_k)$, i.e., the sequence of just the first k elements.

In Fig.14, The structure incident matrix of Petri net model is

$$A=\begin{bmatrix}0&1&1&0\\0&0&0&1\\0&0&0&1\\0&0&0&0\end{bmatrix}\quad B=\begin{bmatrix}0&\frac{1}{2}&\frac{1}{2}&0\\0&0&0&1\\0&0&0&1\\0&0&0&0\end{bmatrix}$$

The structure fitness is $1-\sqrt{(\frac{1}{2})^2+(\frac{1}{2})^2}\Big/4=0.823$, its behavior appropriateness are: behavior precision is 1, behavior recall is 1. In the literature [20], the three indexes are all 1, so the two models are complete the same, but their behavior are not the same, obviously, the measure methods have some difficult to differentiate the two models, our methods can differentiate them.

In Fig.15, The structure incident matrix of Petri net model is

$$A=\begin{bmatrix}0&1&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\\0&0&0&0&0&1\\0&0&0&0&0&0\end{bmatrix}\quad B=\begin{bmatrix}0&1&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\\0&0&0&1&0&1\\0&0&0&0&0&0\end{bmatrix}$$

The structure fitness is 1-1/7=0.857, its behavior appropriateness are: behavior precision is 1, behavior recall is 1.Using the methods In the literature [20], the three indexes are all 1, so the two models are complete the same, but their behavior are not the same.

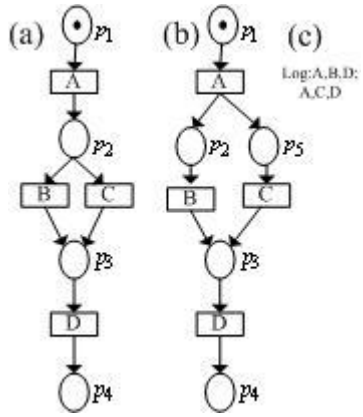


Fig.14. two models (a) and (b) which can accept the log (c).

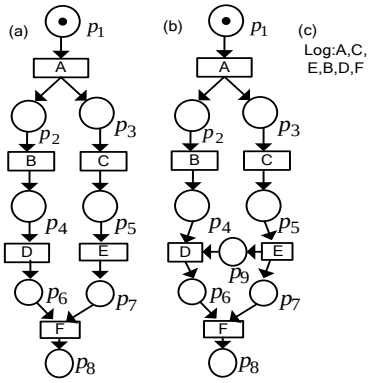


Fig.15. two models (a) and (b) which can accept the log (c).

5.2. Building K-bounded process behavior model based on the induction information

According to the motivating examples and the research framework, the business process mining is important. If we adopt the existing mining methods, it is difficult to mine the exact model which is behavior conformance with the theoretical composite model. Since each mining algorithm has its limitations, it is necessary to find new methods to solve this problem. For example, α-series algorithms can only mine 1- bounded nets, but the composite business process may not be 1- bounded. The region-based mining methods can mine K-bounded (k>1) model, but it is difficult to deal with the behavior dependent relationships.

Due to the particularity of the problem, we analyze the behavior conformance between the mined models with the theoretical composite

model. In the analysis process, we can obtain some behavior dependent information and K - bounded value by analyzing the theoretical composite model in advance, which can induced the business process mining.

In process mining, we can possibly apply the K - bounded value and behavior dependent relationships as induction information to induce the process behavior model's generation.

When K = 1, we propose a business process mining algorithm based behavior dependent relationships, it is showed in algorithm 3. We mainly consider behavior dependent relationships in process mining, and build the optimized model according to the behavior conformance metrics of the Petri nets model.

When K > 1, it is difficult to control the bounded value by adopting the general mining methods, the region-based mining method [22] is a better method, but the method can tackle the behavior dependent relationship well. An improved K-bounded process mining methods is proposed in algorithm 4.

Algorithm3:1-bounded process mining algorithm based on behavior dependent relationships

Input: log traces, induction behavior dependent relationships.

Output: the Petri net model of business process.

(1) Log sequence must be pretreated first; mostly the same sequences will be merged. Then analyzing the direct dependent relationships between all tasks (or transition);

(2) Building the initial model λ_0, according to the pretreated sequences, using the direct dependent relationships to build the initial Petri net model. In building the model, we do not consider the metrics of behavior conformance and other factors, only consider the behavior dependent relationships of tasks (transition), the built model is called as λ_0;

(3) Checking whether the initial model λ_0 accepts the sequence, if some sequences can be accepted, then learn the typical behavior sequences based on the frequency of sequences (i.e. the behavior sequences of large frequency should be accepted), obtaining the new model by learning process;

(4) According to the definition of structure fitness, behavior precision and behavior recall, computing the metrics value of model λ_0 and model λ based on the log sequences, if （ $(fitness(\lambda, L) \geq fitness(\lambda_0, L)$ and $precision(\lambda, \lambda_0, L) \geq \text{Delta1}$ ） or （ $fitness(\lambda, L) \geq fitness(\lambda_0, L)$ and $recall(\lambda, \lambda_0, L) \geq \text{Delta2}$）, then return to step（5）; else, the initial model λ_0 keeps unchanging, repeat the step(3), continue to learn and select the next typical behavior sequences, the selecting consider the different structures with the same sequences, the basic Petri net structure with the same firing sequences can be referenced.

(5) If $recall(\lambda, \lambda_0, L) \geq precision(\lambda, \lambda_0, L)$ then $\lambda_0 = \lambda$, return to the step（6）; else, computing the redundancy degree RED(λ_0) and RED(λ)of the

model λ_0 and the model λ respectively, if $\frac{fitness(\lambda, L)}{\mathrm{RED}(\lambda)} \geq \frac{fitness(\lambda_0, L)}{\mathrm{RED}(\lambda_0)}$ then $\lambda_0 = \lambda$, return to the step（6）; else, determine whether the model λ has the indirect dependent relationship of tasks (transitions), if there exist the indirect dependent relationship, the learning process by the basic structure of indirect dependent relationship, and return to the step (4);

(6) If $fitness(\lambda_0, L) \geq Delta3$ && $\frac{fitness(\lambda_0, L)}{\mathrm{RED}(\lambda)} \geq \mathrm{Delta}$, then the learning process end, and the maximum possibility is λ; else, return to the step（4） and continue to learn, until the learning process end.

(7) Output the model λ.

Algorithm4: K-bounded (K>1) process mining algorithm based on induction information

Input: log traces, induction behavior dependent relationships.

Output: the Petri net model of business process.

(1) Transforming the log traces into the corresponding transition system, then analyzing the direct dependent relationships between all tasks (or transition);

(2) Using the Generate Minimal Regions algorithm [22] to generate the minimal region R of the TS;

(3) Let $TS = (S, E, A, s_{in})$, $PN = (P, T; F, M_0)$, for each $r \in R$, each r is corresponding to a place p, $p_i = r_i$, $|p| = |r|$;

(4) For each $r \in R$, $M_0[r] = r(s_{in})$, each e is corresponding to a place t, $t_i = e_i$, $|t| = |e|$;

(5) Let $ER(e) = \{s \mid \exists s' : (s, e, s') \in E\}$, $SR(e) = \{s \mid \exists s' : (s', e, s) \in E\}$, if $ER(e) \subseteq r$, then linking a directed arc from the p which corresponding to r to t which corresponding to e. if $SR(e) \subseteq r$, then linking a directed arc from the t which corresponding to e to the p which corresponding to r. So, a Petri net model can be obtained, and the model is called as λ_0;

(6) Using the induction behavior dependent relationships, selecting one of dependent relationship to adjust the model λ_0, and obtaining the new model λ;

(7) Computing the metrics value of model λ_0 and model λ based on the log sequences, if （ ($fitness(\lambda, L) \geq fitness(\lambda_0, L)$ and $precision(\lambda, \lambda_0, L) \geq \mathrm{Delta1}$ ） or （ $fitness(\lambda, L) \geq fitness(\lambda_0, L)$ and $recall(\lambda, \lambda_0, L) \geq \mathrm{Delta2}$ ）, then $\lambda_0 = \lambda$; else, the initial model λ_0 keeps unchanging, then considering the next induction information;

(8) If the induction information has been learned end or the model λ is not change, then the modeling process end, and output the model λ .

6. Simulation Experiments

Based on the methods proposed in the paper, we use the ProM framework to analyze the behavior trustworthiness of component interaction. The ProM framework [5] is an open-source tool and it can be downloaded at www.processmining.org. , specially tailored to support the development of process mining plug-ins. In ProM, plug-ins can be categorized. The plug-ins based on data in the event log only is called discovery plug-ins because they do not use any existing information about deployed models. The plug-ins that checks how much the data in the event log matches the prescribed behavior in the deployed models is called conformance plug-ins. Finally, the plug-ins that needs both a model and its logs to discover information that will enhance this model are called extension plug-ins. In our methods, we develop (i) behavior relativity plug-ins to analyze the behavior relativity of interaction components (in Fig.16), (ii) behavior conformance plug-ins to analyze conformance between the building model and the theoretical model (in Fig.17), and (iii) extension plug-ins to use the genetic process mining methods to optimize the process, and to develop performance analyzing tool to compare different process mining methods.

Experiment environment: CPU is Intel dual 3.2 0GHz, Memory is2.00GB, and operation system is Windows XP. Our plus-ins are developed based on the ProM with version 5.2.

Test 1. For the same event logs from the large Benchmarks, giving some known induction information, using our methods and genetic methods presented in the literature [20] respectively, we compare the three indexes including cost, fitness and appropriateness, in order to analyze the relationships between the mined models with the logs by two process mining methods.

We compare our methods with genetic process mining methods which are studied in literature [20]. In Table 1, with the case increase, the costs of two methods are both increasing quickly, but our methods is better than the methods of genetic process mining. The reason is that our methods have smaller solution space because of using the basic behavior structure. The fitness and appropriateness of two methods are irrelevant with the case number, but our methods are better than the genetic process mining methods.

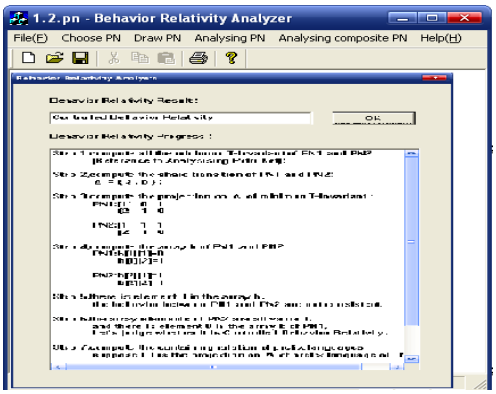


Fig.16. A screenshot of behavior relativity analysis

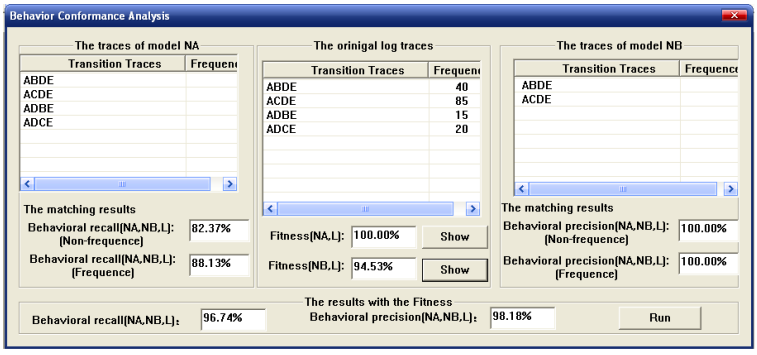


Fig.17. A screenshot of behavior conformance analysis

Table 1. The results of two methods based on large Benchmarks

Log / Index		Log5-1		Log5-2		Log5-3		Log5-4		Log5-5	
		Case	K	Case	K	Case	K	Case	K	Case	K
		100	1	300	1	500	1	700	2	900	2
Our Method	Cost	5.32s		7.68s		10.57s		14.34s		17.84s	
	Fitness	1		1		0.992		0.981		0.958	
	App.	0.996		0.993		0.985		0.972		0.963	
Genetic Method	Cost	5.74s		8.07		11.46s		16.17 s		21.90s	
	Fitness	1		0.997		0.982		0.893		0.887	
	App.	0.994		0.989		0.970		0.882		0.876	

Test 2. For the component cases from the large Benchmarks, giving some known induction information, such as the K-bounded value and some dependent relationships, using our methods and genetic methods presented in the literature [20] respectively, we compare the some indexes using two methods under different interacting business process.

Table 2. The result of two methods based on benchmarks

Index / methods	Interacting BP	(Bp1, Bp2)	(Bp1, Bp3)	(Bp1, Bp4)	(Bp1, Bp5)	(Bp2, Bp3)	(Bp2, Bp4)
	K-bounded	1	1	2	2	2	2
Our Methods	K-bounded	1	1	2	2	2	2
	Structure Fitness	1	1	1	1	0.996	0.994
	Behavior Precision	1	1	0.985	0.978	0.971	0.963
	Behavior Recall	0.992	0.986	0.978	0.966	0.959	0.952
Genetic Methods	K-bounded	1	1	1	1	1	1
	Structure Fitness	1	1	0.993	0.989	0.984	0.977
	Behavior Precision	0.996	0.994	0.975	0.958	0.920	0.901
	Behavior Recall	0.969	0.971	0.937	0.915	0.895	0.886

In Table 2, a tuple (Bpi, Bpj) represents the interaction between the business Bpi and the business process Bpj. By analyzing, the K-bounded value is given. The experiment results show that our methods are better than the genetic mining methods, especially, when the K-bounded value of composite business process is larger than 1, the difference is larger.

7. Conclusions

At present, the trustworthiness of business process is a study focus, the essence characteristics of trustworthy business process are the execution result and behavior can be predictable. Under the open and dynamic changing environment, some complex demands need several business processes interacting to implement, these cause the composite business process to behave uncontrollably, uncertainly, and unpredictably. The paper presents a behavior trustworthiness analysis method of business process based on induction information. Firstly, aimed to the internal factors, we analyze the consistent behavior relativity in order to guarantee the predictable function of business process. Then, for the external factors, in order to analyze the behavior change of business process, we propose a process mining methods based on induction information. Finally, using methods above mentioned, we test our methods in large Benchmark, and compare our methods with genetic process mining. Theoretical analysis and

experimental results indicate that our method is better than the methods of genetic process mining.

Based on the theoretical analysis and experimental results, the innovation and advantage of the paper are: 1) In order to analyze the predictable interaction function, consistent behavior relativity based on Petri net is presented. 2) An effective measure methods of behavior conformance is proposed, the methods can solve these problems such an Fig.14 and Fig.15, but these problems can not solved in literature in[20]. 3) Business process mining methods based on induction information is proposed, which can take into account induction information from the theoretical composite model, and avoid the blindness of building process model.

In the future, we would like to study the trustworthy evaluation of networked software, and study the adaptation methods of non-consistent behavior relativity. Moreover, it is also one of our future works to study the intelligence and dynamic behavior analyzing methods of complex business process.

Acknowledgement. We would like to thank the support of the National Natural Science Foundation of China under Grant No.90818023, No.60873144 and No.60973050, the Program for Changjiang Scholars and Innovative Research Team in University, the Major State Basic Research Development Program of China under Grant No. 2010CB328101, the Natural Science Foundation of Educational Committee of Anhui Province of China (KJ2011A086, KJ2010B310and KJ2009A50), the Natural Science Foundation of Shanxi (Nos. 2011021013-2).

References

1. Zheng Z.M., Ma S.L.: Dynamical characteristics of software trustworthiness and their evolutionary complexity. Science in China Ser F-Inf Sci, 52(8), 1328–1334.(2009)
2. Wang H.M., Tang Y.B., Ying G.: The trustworthy mechanism of internet software. Chinese Science (E series), 36(10), 1156－1169. (2006)
3. Liu K, Shan Z.G., Wang J.: The general study on the major project of trustworthy software. Chinese Science Foundation, 22(3), 145-151. (2008)
4. W.M.P. van der Aalst, T. Weijters and L. Maruster, Workflow Mining: Discovering process models from event logs, IEEE Transactions on Knowledge and Data Engineering. 16(9), 1128-1142. (2004)
5. W.M.P.van der Aalst, B.F.van Dongen, J.Herbst, L.Maruster and G. Schimm, Workflow Mining: A Survey of Issues and Approaches, Data and Knowledge Engineering. 47(2), 237-267. (2003)
6. W.M.P.van der Aalst, H.A. Reijers and A.J.M.M. Werjters, Business process mining: An industrial application, Information System.vol 32 ,713−732. (2007)
7. J. Koehler, G. Tirenni, and S. Kumaran. From business process model to consistent implementation: A case for formal verification methods. In Proc. 6th IEEE International Enterprise Distributed Object Computing Conference (EDOC), pages 96–106, 2002.

8. E. Kindler and T. Vesper. A temporal logic for events and states. In Proc. 19th International Conference on Application and Theory of Petri Nets, volume 1420 of Lecture Notes in Computer Science, pages 365–384. Springer, 1998.
9. C. Klauck and H.-J.Mueller. Formal business process engineering based on grammar graphs. International Journal on Production Economics, 50:129–140, 1997.
10. J. J¨urjens. Towards development of secure systems using UML. In Proc. Fundamental Approaches for Software Engineering (FASE), pages 187–200, 2001.
11. T. Lodderstedt, D. Basin, and J. Doser. Secure UML: A UML-based modeling language for model-driven security. In Proc. 5th International Conference on the Unified Modeling Language, volume 2460 of Lecture Notes in Computer Science, pages 425–441. Springer, 2002.
12. Du YY, Jiang CJ, Zhou MC, et al. Modeling and monitoring of E-commerce workflows. INFORMATION SCIENCES, 179(7):995-1006. (2009)
13. Michael Backes, Birgit Pfitzmann, Michael Waidner. Security in Business Process Engineering, LCNS 2678, pages 168 - 183, June 2003.
14. Hassan T , Morteza A , Rasool J . Trust-based user-role assignment in role-based access control//Proceedings of the AICCSA. Amman , Jordan , 807-814.(2007)
15. Ping Sun, Changjun Jiang. Analysis of workflow dynamic changes based on Petri net. Information and Software Technology, v 51, n 2, p 284-292, February 2009
16. Montali, M.: Specification and verification of declarative open interaction models: A logic-based approach. Lecture Notes in Business Information Processing, v 56 LNBI, p 1-402, (2010)
17. Alam, Masoom ,Nauman, Mohammad; Zhang, Xinwen: Behavioral attestation for web services based business processes. International Journal of Web Services Research, v 7, n 3, p 52-72, (2010)
18. Alam, Masoom Nauman, Mohammad; Zhang, Xinwen; Ali, Tamleek; Hung, Patrick C.K. Behavioral attestation for business processes. 2009 IEEE International Conference on Web Services, ICWS 2009, p 343-350, (2009)
19. Wan, Kaiyu (East China Normal University, Shanghai, China); Muhammad, Mubarak; Alagar, Vasu, A formal model of business application integration from Web services. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), v 5404 LNCS, p 656-667, (2009)
20. A.K. Alves de Medeiros, W.M.P. van ,A.J.M.M. Weijters, Quantifying process equivalence based on observed behavior, Data & knowledge engineering, 64,55–74 (2008)
21. K. A. de Medeiros, A. J. M. M. Weijters and W. M. P. van der Aalst, Genetic process mining: an experimental evaluation, Data Mining and Knowledge Discovery. 14,245–304. （2007）
22. J. Carmona, J. Cortadella and M. Kishinevsky,New Region-Based Algorithms for Deriving Bounded Petri Nets, IEEE Transactions on Computers. 59(3),371-384. (2010)
23. T. Murata, Petri nets: Properties, analysis and applications. Proc. IEEE, 77(4) (1989) 541–580.
24. C.J. JIANG, Behavior theory and its application of Petri net, (Higher education press, 2003).

25. Wen Lijie, Wil M. P. van der Aalst and Wang Jianmin etc, Mining Process Models with Non-Free-Choice constructs, Data Mining and Knowledge Discovery. 15(2), 145-180.(2007)
26. Rozinat and W.M.P. van der Aalst, Conformance Testing: Measuring the Fit and Appropriateness of Event Logs and Process Models, Workshop Business Process Intelligence (BPI '05). vol1,1-12. (2005)

Xianwen Fang received his M.A. degree from Shandong University of Science and Technology, China, in 2004, and PhD. degree in the key Lab of Service Computing at Tongji University in 2011. He has been an associate processor with Anhui University of Science and Technology, China, since 2007. His research interests include Petri net, trustworthy software and Web services.

Changjun Jiang received the PhD. degree from Chinese Academy of Sciences, Beijing, China, in 1995. He is currently a Professor with the Department of Computer Science and Engineering, Tongji University, China. His current areas of research are concurrent theory, Petri net, and formal verification of software. He has published more than 100 papers in domestic and international academic journals and conference proceedings. These papers are embodied more than 150 times by SCI and EI and are cited more than 200 times by others.

Zhixiang Yin received her M.A. degree from Nanjing Normal University of China in 1996, and PhD. degree in the Huazhong University of Science and Technology in 2003. His research interests include Petri net and trusted software.

Xiaoqin Fan received her M.A. degree from TaiYuan University of Science and Technology, China, in 2002, and PhD. degree in the key Lab of Service Computing at Tongji University in 2010. Her research interests include Petri net and Web services.

Received: April 11, 2010; Accepted: January 19, 2011.

DOI: 10.2298/CSIS100906032Z

Voice Activity Detection Method Based on Multi-valued Coarse-graining Lempel-Ziv Complexity

Huan Zhao[1], Gangjin Wang[1], Cheng Xu[1], and Fei Yu[2]

[1] School of Information Science and Engineering, Hunan University,
410082 Changsha, P. R. China
hzhao@hnu.edu.cn, wgangjin@163.com, chengxu@public.cs.hn.cn
[2] Jiangsu Provincial Key Laboratory of Computer Information Processing Technology,
215000 Suzhou, P. R. China
hunanyufei@126.com

Abstract. One of the key issues in practical speech processing is to locate precisely endpoints of the input utterance to be free of non-speech regions. Although lots of studies have been performed to solve this problem, the operation of existing voice activity detection (VAD) algorithms is still far away from ideal. This paper proposes a novel robust feature for VAD method that is based on multi-valued coarse-graining Lempel-Ziv Complexity (MLZC), which is an improved algorithm of the binary coarse-graining Lempel-Ziv Complexity (BLZC). In addition, we use fuzzy *c*-Means clustering algorithm and the Bayesian information criterion algorithm to estimate the thresholds of the MLZC characteristic, and adopt the dual-thresholds method for VAD. Experimental results on the TIMIT continuous speech database show that at low SNR environments, the detection performance of the proposed MLZC method is superior to the VAD in GSM ARM, G.729 and BLZC method.

Keywords: speech processing, voice activity detection, Lempel-Ziv complexity, multi-valued coarse-graining, fuzzy *c*-Means clustering algorithm, Bayesian information criterion algorithm.

1. Introduction

Voice activity detection (VAD) is used to distinguish speech from noise and is required in many speech applications, such as speech recognition [1], speech enhancement [2], voice biometrics [3], and speech coding [4]. The VAD process is demonstrated in Fig. 1 [5]. Effective VAD of speech signals can not only reduce the amount of speech signals processing operations, but also improve system performance effectively.

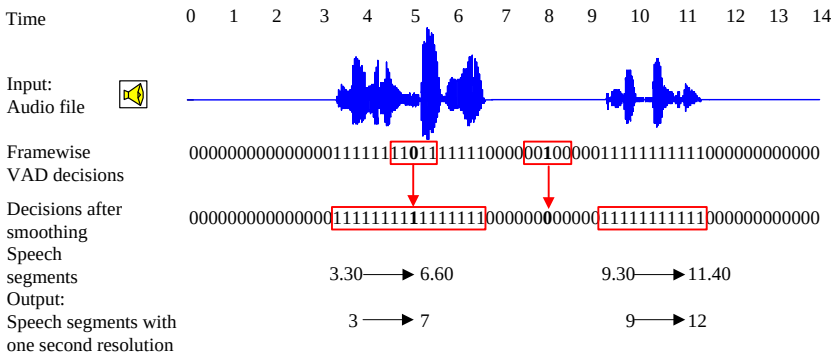


Fig. 1. Illustration of VAD process [5]

At present, various VAD algorithms have been proposed, such as Itakura LPC distance measure [6], cepstral features [7], energy levels [8], the difference of energy and zero-crossing rate [9], spectral entropy [10], energy-spectral entropy [11] and distance entropy [12]. In the condition of high SNR, the differences between voices and background noises are rather distinct so that many algorithms have good performances in VAD. However, the present VAD algorithms have the problem of detection performance in low SNR environments, especially in the presence of non-stationary noise.

From a physics and mathematics viewpoint, many studies have shown that voice signal has non-linear and non-stationary features [13]. Relevant aspects of current research applied in VAD are: permutation entropy [14], approximate entropy [15], C_0 complexity [16], Lempel-Ziv complexity (LZC) [17] and so on. Among them, Ref. [17] proposed a novel VAD algorithm based on binary coarse-graining Lempel-Ziv complexity (BLZC) in the white Gaussian noise environment with a good detection result, but its detection performance declines dramatically in the low SNR non-Gaussian and non-stationary noise environments (such as: factory noise, babble noise, etc.). Considering that the binary coarse-graining method may lose some important information on dynamical systems, we present a novel VAD method based on multi-valued coarse-graining LZC (MLZC). Besides, we use fuzzy *c*-Means clustering (FCMC) [18, 19] and the Bayesian information criterion (BIC) algorithm [19, 20] to estimate the thresholds of MLZC and via dual-thresholds method for VAD. Experimental results show that in a variety of noisy environments, MLZC has a better detection performance than the VAD in GSM ARM, G.729 and BLZC method.

This paper is organized as follows. In Sec. 2, the multi-valued coarse-graining Lempel-Ziv complexity (MLZC) feature is described. Next, the FCMC algorithm and BIC algorithm are applied to estimate the thresholds of the MLZC feature and dual-thresholds VAD method are given in Sec. 3. In Sec. 4, simulations are provided to verify the MLZC approach whose results are compared with the VAD in GSM ARM, G.729 and BLZC method. Finally, the conclusion and further researches are given in Sec. 5.

2. Multi-valued Coarse-graining Lempel-Ziv Complexity

With the development of science, especially of nonlinear science, a common viewpoint has been formed, that is, the speech signal is a complex time series and acts as an unstable strange attractor in a chaotic system rather than a random signal. There have been many definitions of complexity measure, for example, Kolmogorov Complexity (KC). Lempel and Ziv introduced an easy mathematical method to calculate the measure of Kolmogorov Complexity which is defined as Lempel-Ziv complexity (LZC) [21]. LZC analysis is based on a coarse-graining of measurements, i.e. the signal to be analyzed is transformed into a sequence whose elements are only a few symbols. The most widely calculation of LZC is based on the binary sequence which generated by the mean value or zero of the input signal, but the binary sequence cannot well characterize speech signal and may lose some important speech information easily. Therefore, we present a novel VAD method based on multi-valued coarse-graining LZC (MLZC). In the following section we present a detailed study of the MLZC for VAD.

2.1. Binary Coarse-graining Method

In nonlinear time series, the traditional computation of complexity is based on binary sequence, i.e. given a dynamic system time sequence $X=\{x_i|i=1,2,\ldots,n\}$, and then the average for the time series is

$$X_{ave}=\frac{1}{n}\sum_{i=1}^{n}x_i\ , \tag{1}$$

so, the binary sequence $s_i(1\le i\le n)$ can be obtained by

$$s_i=\begin{cases}1, & \text{if} \quad x_i>X_{ave}\\ 0, & \text{else}\end{cases}. \tag{2}$$

2.2. Multi-valued Coarse-graining Method

As the time series produced by the binary coarse-graining method is likely to be missing some important information on dynamical systems, we use multi-valued coarse-graining method reconstruct the time series, which is defined as follows [22]:

Let $X=\{x_i|i=1,2,\ldots,n\}$ be a set of time sequences, and let $x_{\max}$ be the maximum value and $x_{\min}$ be the minimum value of the set. Besides, let $L(L>2)$ be the data coarse-graining segment number of the set, then we define

$$d=\left(x_{\max}-x_{\min}\right)/L\ . \tag{3}$$

Let $\{y_j|j=1,2,\ldots,L\}$be a set of symbol sequences, where each value of y_j is different. Let $s_i(1\le i\le n)$ be the symbol string which is the L-valued coarse-graining result of the set X, defined by

$$s_i = \begin{cases} y_j, \ x_{\min} + (j-1)d \le x_i \le x_{\min} + jd \ \ (j = 1,2,\cdots,L) \\ y_L, \ x_i = x_{\max} \end{cases}. \tag{4}$$

2.3. Lempel-Ziv Complexity

In the previous section we got the symbol string $P=\{s_i|i=1,2,\ldots,n\}$, and the basic idea of Lempel-Ziv complexity analysis is as follows [23, 24]:

Let S and Q denote, respectively, subsequence of the sequence P and SQ be the concatenation of S and Q, while sequence SQv is derived from SQ after its last character is deleted (v means the operation to delete the last character in the sequence). Let $V(SQv)$ denote the vocabulary of all different subsequences of SQv. First, let $c(n)=1$, $S=s_1$, $Q=s_2$, therefore, $SQv=s_1$. Now, suppose $S=\{s_1,s_2,...,s_r\}$, $Q=s_{r+1}$. If $Q\in V(SQv)$, then s_{r+1} is a subsequence of $\{s_1,s_2,\ldots,s_r\}$. At this point, S needn't change and Q update to be $Q=\{s_{r+1},s_{r+2}\}$, then judge whether Q belongs to SQv or not (meanwhile, S needn't change, Q updated and SQv should also update), and continue until $Q\notin V(SQv)$. Now, suppose $Q=\{s_{r+1},s_{r+2},\ldots,s_{r+l}\}$, then $\{s_{r+1},s_{r+2},\ldots,s_{r+l}\}$ is not a subsequence of $\{s_1,s_2,\ldots,s_r,s_{r+1},\ldots,s_{r+l-1}\}$, so increase $c(n)$ by one. Afterwards, combine S with Q and S is renewed to be $S=\{s_1,s_2,\ldots,s_r,s_{r+1},s_{r+2},s_{r+3},\ldots,s_{r+l}\}$, by this time take Q as $Q=s_{r+l+1}$. Repeat these procedures until Q is the last character. At this time, the number of different subsequences is $c(n)$. If the length of the symbol sequence is n, the upper bound of $c(n)$ is given by

$$c(n) < \frac{n}{(1-\varepsilon_n)\log(n)}, \tag{5}$$

where, n is a small quantity and $\varepsilon_n\to 0$ ($n\to 0$). Therefore, in general $n/\log(n)$ is upper bound of $c(n)$, i.e.,

$$\lim_{n\to\infty} c(n) = b(n) \equiv \frac{n}{\log(n)}, \tag{6}$$

so, $c(n)$ is the asymptotic behavior of the random sequence, and $c(n)$ can be normalized via this limit

$$C(n) = \frac{c(n)}{b(n)}. \tag{7}$$

Example: In order to make the calculation of the $c(n)$ easily understood, Fig. 2 shows how to transform a segment of a speech signal series into a ternary sequence by three-valued coarse-graining method (i.e. $L=3$) and the result of complexity analysis on the ternary sequence. After three-valued coarse-graining, the resulting P=000001000012111122221211 (length n=25), and the complex counter $c(n)$ of the sequence P is calculated by complexity

analysis as follows. Symbol "•" denotes the end of each different subsequence, and the number of "•" is equal to the value of $c(n)$.

1) Fist character (i.e. in this case 0) is always a novel one. Therefore, the first subsequence is →0•, i.e. $c(n)$=1.

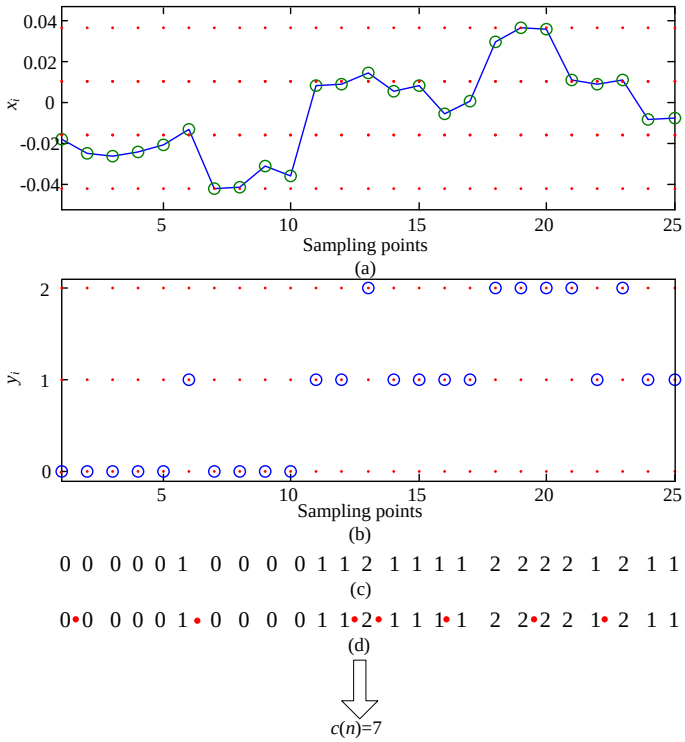


Fig. 2. An illustration showing how to calculate the $c(n)$ (L=3). (a) A segments of speech signal series, (b) Transform the speech signal series into a ternary sequence by three-valued coarse-graining method, (c) Ternary sequences, (d) Complexity analysis result

2) The second character of P is 0 and this is the first subsequence. In this situation, old subsequence S=0, the current subsequence Q=0, concatenated subsequence SQ=00 and previous subsequence SQv=0. Therefore, $Q \in SQv$, so Q is not a new subsequence →0•0, i.e. $c(n)$=1.

3) The third character of P is still 0. The old subsequence (before "•") S=0, the current subsequence Q=00, concatenated subsequence SQ=000 and previous subsequence SQv=00. Therefore, $Q \in SQv$, so Q is not a new subsequence →0•00, i.e. $c(n)$=1.

4) Results of the fourth and fifth character are the same as the third one. When came to the sixth character of P is 1, the old subsequence S=0, the current subsequence Q=00001, concatenated subsequence SQ=000001 and

previous subsequence *SQv*=00000. Therefore, $Q \notin SQv$, so *Q* is a new subsequence →0•00001•, i.e. *c*(*n*)=2.

5) The seventh character of *P* is 0. The old subsequence *S*=000001, the current subsequence *Q*=0, concatenated subsequence *SQ*=0000010 and previous subsequence *SQv*=000001. Therefore, $Q \in SQv$, so *Q* is not a new subsequence →0•00001•0, i.e. *c*(*n*)=2.

6) Before the 12th character of *P*, the new subsequence has not appeared. When came to the 12th character of *P* is 1, the old subsequence *S*=000001, the current subsequence *Q*=000011, concatenated subsequence *SQ*=000001000011 and previous subsequence *SQv*=00000100001. Thus, $Q \notin SQv$, so *Q* is a new subsequence →0•00001•000011•, i.e. *c*(*n*)=3.

7) The 13th character of *P* is 2. The old subsequence *S*=000001000011, the current subsequence *Q*=2, concatenated subsequence *SQ*=00000100001 12 and previous subsequence *SQv*=000001000011. Therefore, $Q \notin SQv$, so *Q* is a new subsequence →0•00001•000011•2•, i.e. *c*(*n*)=4.

8) The 14th character of *P* is 1. The old subsequence *S*=0000010000112, the current subsequence *Q*=1, concatenated subsequence *SQ*=00000100001 121 and previous subsequence *SQv*=0000010000112. Therefore, $Q \in SQv$, so *Q* is not a new subsequence →0•00001•000011•2•1, i.e. *c*(*n*)=4.

9) When came to the 16th character of *P* is 1. The old subsequence *S*=0000010000112, the current subsequence *Q*=111, concatenated subsequence *SQ*=0000010000112111 and previous subsequence *SQv*=000001000011211. Therefore, $Q \notin SQv$, so *Q* is a new subsequence →0•00001•000011•2•111•, i.e. *c*(*n*)=5.

By this process, the sequence *P* is scanned and partitioned as follows:

P=0•00001•000011•2•111•122•221•211

The number of symbol "•" in *P* is seven and this is the value of complexity counter *c*(*n*).

2.4. Algorithm Validation

To verify the effectiveness of LZC in detecting the nonlinear signal, the Logistic model was adopted as a verification object. The Logistic map is a simple mathematical model that describes how the quantity changes of insects over time, which is the best known of the nonlinear dynamic system. This is a one-dimensional Logistic map defined by [25]

$$x_{n+1} = \lambda x_n \left(1 - x_n\right), \tag{8}$$

where λ is an external parameter, $1 \le \lambda \le 4$, and the range of x_n is changed from a circle to the interval [0,1]. Fig. 3(a) shows the evolution of Logistic map bifurcation diagram in the range $3.5 < \lambda < 4$. It is known that there is a stable fixed point $x_n=0$ in the range $0 \le \lambda < 1$, and another stable fixed point $x_n = 1-1/\lambda$ in the range $1 \le \lambda < 3$, we call this periodic be the 1-cycle; when $3 \le \lambda < 1+\sqrt{6}$, x_n always oscillates between two values, and the two values are dependent on λ, we call this periodic be the 2-cycle; when $1+\sqrt{6} \le \lambda < 3.545$, x_n always

oscillates between four values, the 2-cycle is repelling, but a 4-cycle; when $3.545 \leq \lambda < 3.56995$, x_n oscillates between 8 values, then 16, 32…, i.e. the 8-cycle, 16-cycle, 32-cycle…; when $3.56995 \leq \lambda < 4$, the time series undergo the four different evolution stages, i.e. fixed point, unstable fixed point, periodic, and chaotic, until the chaos phenomena. Fig. 3 (b)~(c) shows the change of BLZC and MLZC under the Logistic map evolution, respectively.

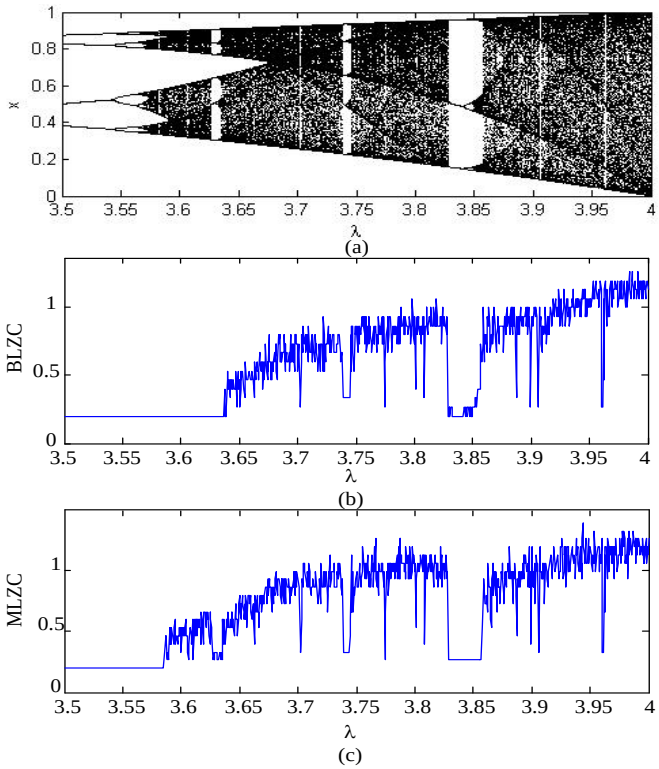


Fig.3. Logistic equations and LZCs for varying control parameter. (a) Bifurcation diagram, (b) BLZC, (c) MLZC

As shown in Fig. 3 (c), when the time series bifurcate evolutes from one state to another state, the MLZC changes obviously, with the same change paces of the Logistic map evolution, i.e. in the range $3.56 < \lambda < 4$. However, Fig. 3 (b) shows that the changes of the BLZC occurs in the range $3.64 < \lambda < 4$. Therefore, the MLZC is superior to the BLZC, which can detect and amplify small changes in the time series and can be used to detect mutations in the signal. Figure 4 displays the LZC of the clean speech under the different coarse-graining methods. We see that the BLZC feature is difficult to distinguish between voice and silence, while $L>2$, different LZC under the L can accurately characterize voice and silence. Without loss of generality, we take $L=3$ in the following discussion.

3. Thresholds Estimation and Algorithm

In this paper, FCMC [18, 19] and BIC [19, 20] algorithms are used to estimate the thresholds of the MLZC feature for VAD. Besides, we use dual-thresholds method for VAD. It can maintain fast tracking speed of environment change

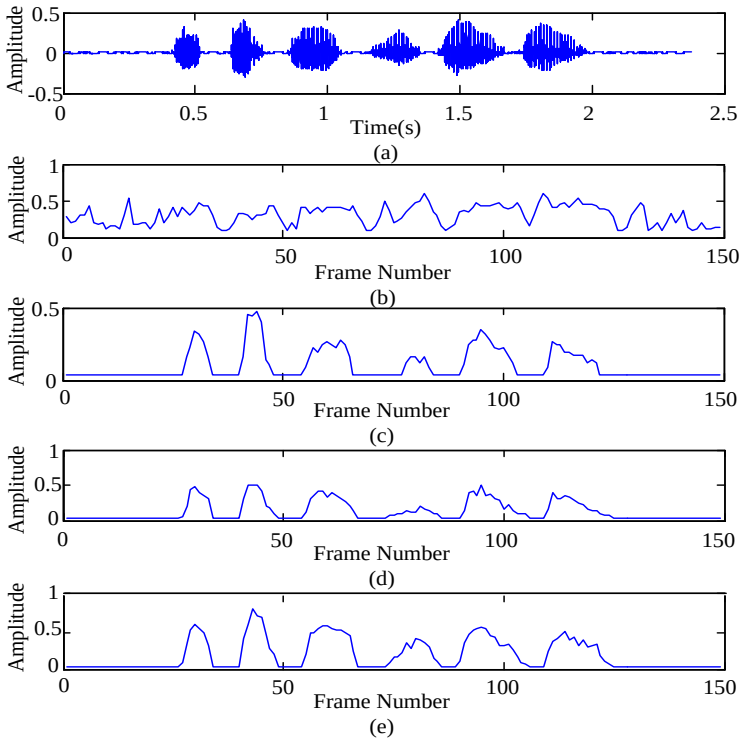


Fig.4. LZC under different coarse-graining methods. (a) Clean speech waveform, (b) BLZC for noisy speech, (c)~(e) MLZC for noisy speech, L=3, 4, 5, respectively

when combined with online update. The brief introduction to the algorithms is given as follows.

3.1. Fuzzy *c*-Means Clustering Algorithm

Assume that the unlabeled object data $X=\{x_i|i=1,2,\ldots,N\}$, C is the expected cluster number and $\{m_j|j=1,2,\ldots,C\}$ are the center of the clusters. The most widely used objective function model for fuzzy c-Means clustering (FCMC) in X is the weighted within groups sum of squared errors objective function J_b, which is used to define the constrained optimization problem:

$$\min\left\{ J_b = \sum_{j=1}^{C}\sum_{i=1}^{N}\left(\mu_j(x_i)\right)^b \left\|x_i - m_j\right\|^2 \right\}, \quad s.t. \sum_{j=1}^{C}\mu_j(x_i) = 1, i = 1,2,\cdots,N \tag{9}$$

where b>1 is the fuzzifier parameter, and $\mu_j(x_i)$ is the grade of membership of x_i in the j-th cluster and subjects to the constrains.

Minimization of J_b subjects to constrains, leads to the following function:

$$m_j = \frac{\sum_{i=1}^{N}\left(\mu_j(x_i)\right)^b x_i}{\sum_{i=1}^{N}\left(\mu_j(x_i)\right)^b},\ j = 1,2,\cdots,C, \tag{10}$$

$$\mu_j(x_i) = \frac{\left(1/\left\|x_i - m_j\right\|^2\right)^{1/(b-1)}}{\sum_{k=1}^{C}\left(1/\left\|x_i - m_k\right\|^2\right)^{1/(b-1)}},\ i = 1,2,\cdots,N,\ j = 1,2,\cdots,C. \tag{11}$$

Using iterative method for solving (10) and (11), we get the fuzzy c-Means clustering algorithm.

3.2. Bayesian Information Criterion Algorithm

For a speech signal, we need to determine whether it contains a clean voice only, or also includes the background noise. In this paper, we use the Bayesian information criterion (BIC) algorithm to determine the best cluster number [19, 20].

According to BIC, the best model number is the one with maximized BIC value. If voice and the background noise are modeled as a multi-variance Gaussian distribution $N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, where $\boldsymbol{\mu}_i$ is the sample mean vector and $\boldsymbol{\Sigma}_i$ is the sample covariance matrix, the BIC value is [19]

$$\mathrm{BIC}(C) = \sum_{i=1}^{C}\left\{-\frac{1}{2}N_i \log\left|\boldsymbol{\Sigma}_i\right|\right\} - \frac{\log(N)}{2}\lambda_p C\left[d + \frac{d(d+1)}{2}\right], \tag{12}$$

where N is the total sample number, N_i is the number of sample in the i-th cluster, λ_p is the penalty weight, and d is the dimension of the feature space.

We applied the BIC criterion to determine the best cluster number C_{best} for VAD can be present as

$$C_{best} = \begin{cases} 1, & \text{if} \quad \mathrm{BIC}(1) > \mathrm{BIC}(2) \\ 2, & \text{else} \end{cases}. \tag{13}$$

3.3. Thresholds Estimation

In this section, how to use the FCMC and BIC algorithms to ascertain the thresholds for VAD is illustrated. Before making thresholds estimation, we need to pass framing, adding window and other pretreatment to speech signals. The algorithm steps are as follows [19]:

Step 1 Calculate the MLZC for each frame by (7).

Step 2 Given the cluster number C=2, making FCMC on the MLZC of frames.

Step 3 Use (13) to determine the best cluster number C_{best}.

Step 4 IF C_{best} =1

Step 2 obtained the cluster center m_{11}, and then the thresholds formula of MLZC is:

$$\mathrm{TH}_{high} = m_{11} + \alpha_{high}, \tag{14}$$

$$\mathrm{TH}_{low} = m_{11} + \alpha_{low}, \tag{15}$$

where TH_{high} and TH_{low} are the higher and lower thresholds respectively, and α_{high}, α_{low} are empirical constants.

ELSE

Step 2 obtained the cluster centers m_{21} and m_{22}, then the mean of MLZC of the voice and background noise are given by

$$m_{speech} = \max\{m_{21}, m_{22}\}, \tag{16}$$

$$m_{noise} = \min\{m_{21}, m_{22}\}. \tag{17}$$

So the threshold formula of MLZC is:

$$\mathrm{TH}_{high} = m_{noise} + (m_{speech} - m_{noise})\beta_{high}, \tag{18}$$

$$\mathrm{TH}_{low} = m_{noise} + (m_{speech} - m_{noise})\beta_{low}, \tag{19}$$

where, β_{high}, β_{low} are empirical constants.

END

3.4. Dual-thresholds Method

After obtaining the thresholds of MLZC by the above steps, we use the dual-thresholds method for VAD. The dual-thresholds arithmetic is first introduced by Lawrence Rabiner [26]. The improved algorithm we use for VAD can be present as follows [12, 27].

Step 1 As shown in Fig. 5, the higher threshold TH_{high} and the lower threshold TH_{low} are got in Sec. 3.3.

Step 2 Comparing the current MLZC with the TH_{high}, if MLZC>TH_{high}, the current frame affirmatively belongs to voice signal segment. As a result, we can obtain the two approximate endpoints N_1 and N_2.

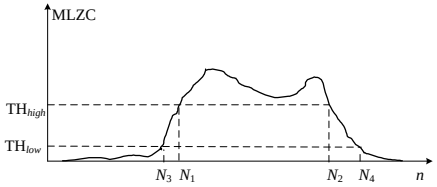


Fig. 5. Dual-thresholds for VAD

Step 3 Searching forward from N_1, if in the first frame MLZC<TH_{low}, then we recorded the frame as N_3. Besides, searching backward from N_2, if in the first frame MLZC<TH_{low}, then we recorded the frame as N_4. Therefore, we can initially get the starting endpoint N_3 and the ending endpoint N_4. N_3N_4 is defined as a segment of voice signal.

Step 4 If the voice segment is less than 4 frames, take it as the result of mutation of the background noise, and it should be omitted.

Step 5 If the interval between adjacent voice segments is less than 0.2s, merge the two adjacent voice segments.

4. Experiments

4.1. Experimental setup

The original speeches used for simulation and test are taken from the DARPA TIMIT Acoustic-Phonetic Speech Corpus [28]. 450 English sentences are selected. All of them are sampled at 16 kHz, and quantized into 16 bits. Different background noises with different time-frequency distributions were taken from the NOISEX-92 database [29]. The tested noisy environments include White noise, Babble noise, Factory noise, Volvo (car) noise. Noise has been added to the clean speech signal with 5 SNRs (0, 5, 10, 15, 20dB).

Based on the above experimental speech environments, we set the values of each parameter as follows: the speech frame length is 512 (32ms); frame shift is 256 (16ms); window function is hamming window; and experiments show that the threshold estimated parameters are: α_{high}=5.4, α_{low}=-0.24, β_{high} =0.15, β_{low}=-0.042.

4.2. Experimental Results of VAD

In this section, we carried out a series of experiments to evaluate the effectiveness of the VAD algorithm. As shown in Fig. 6~Fig. 9, the VAD outputs for the MLZC are investigated in the given speeches, whose results

are compared with the baseline algorithms from GSM AMR VAD [30], G.729 VAD [31] and BLZC [17]. We can see that, as the SNR dropped, GSM AMR's VAD detection performance became stably, and can only detect part of the voice; G.729's VAD detection performance declined sharply, especially when SNR = 5dB (in Fig. 9(c)), the whole noisy speech was detected as the speech sound; BLZC's detection performance has also declined but still been able to find all of the voice, yet falsely detected the three speech segments as one voice segment; However, MLZC's detection performance showed good robustness, and accurately detected the speech starting and ending positions. In order to better distinguish BLZC and MLZC detection performance, Fig. 10 and Fig. 11 respectively show the VAD results of the BLZC and MLZC in different noise environments (SNR = 5dB). As can be seen from the figures, the voice truncated errors (the voice misclassified as the noise) of the BLZC method is very common (such as, in babble noise and factory noise environments) and its extended errors (the noise misclassified as the voice) exit in the 4 noises environments. Fortunately, the MLZC only h-

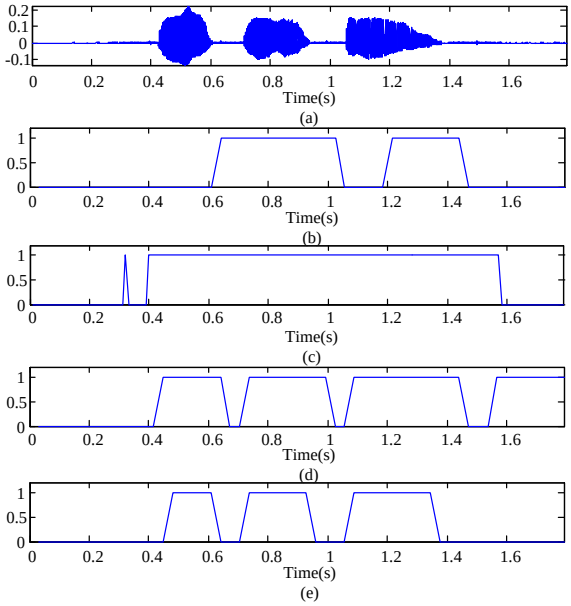


Fig. 6. (a) Clean speech waveform, (b) GSM AMR VAD results, (c) G.729 VAD results, (d) BLZC VAD results, (e) MLZC VAD results

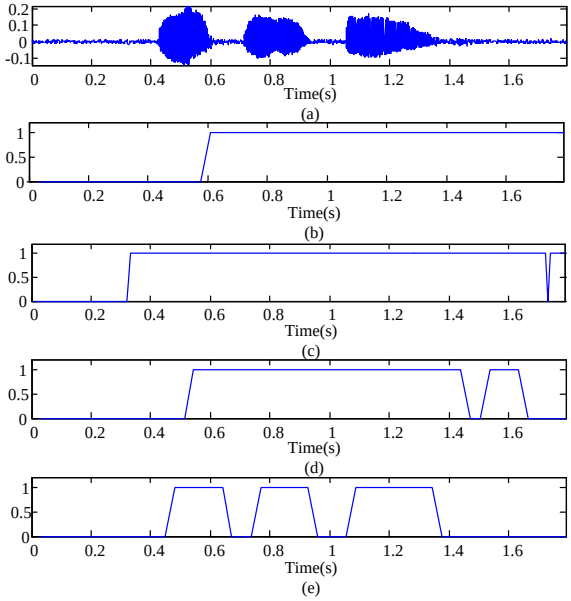


Fig. 7. (a) Noisy speech waveform (Babble noise, SNR=15dB), (b) GSM AMR VAD results, (c) G.729 VAD results, (d) BLZC VAD results, (e) MLZC VAD results

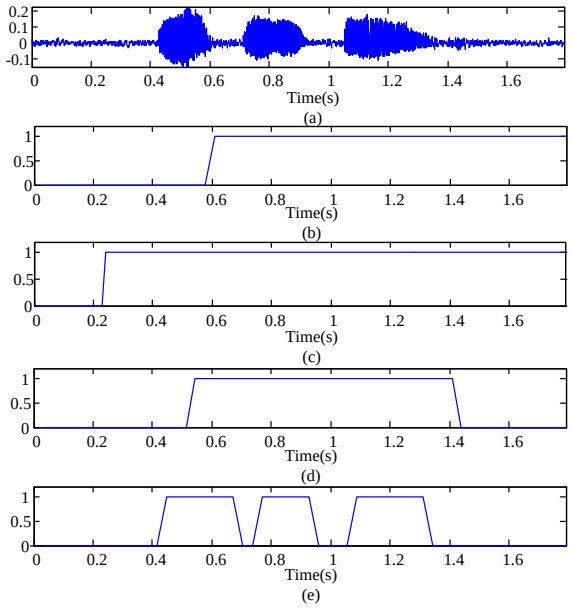


Fig. 8. (a) Noisy speech waveform (Babble noise, SNR=10dB), (b) GSM AMR VAD results, (c) G.729 VAD results, (d) BLZC VAD results, (e) MLZC VAD results

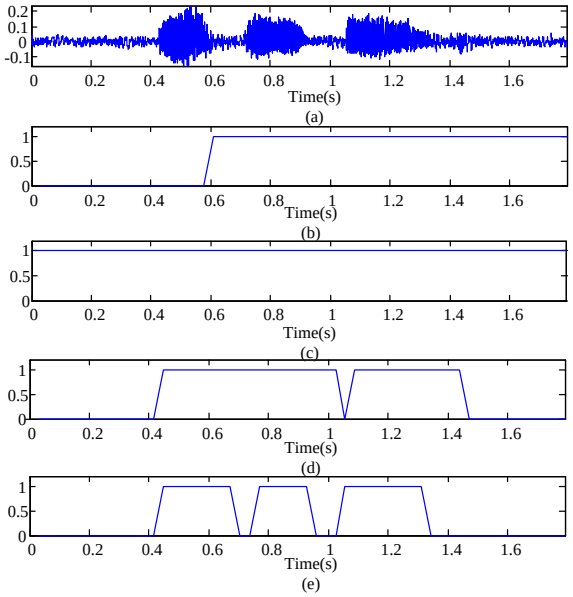


Fig. 9. (a) Noisy speech waveform (Babble noise, SNR=5dB), (b) GSM AMR VAD results, (c) G.729 VAD results, (d) BLZC VAD results, (e) MLZC VAD results

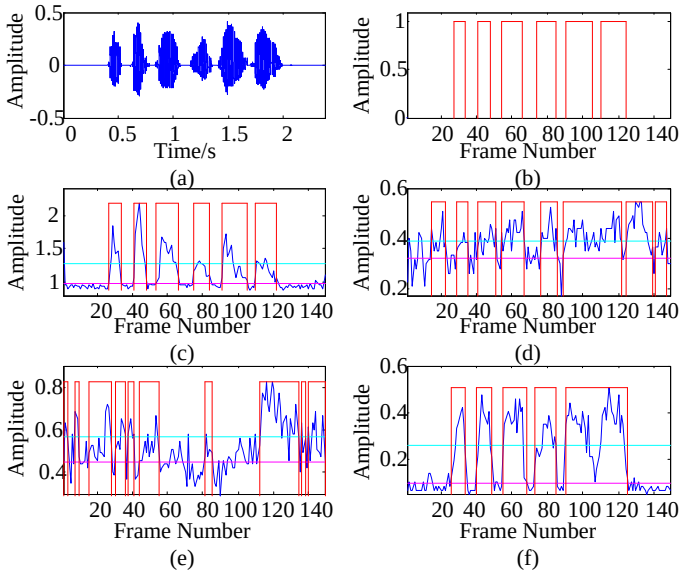


Fig. 10. VAD results by BLZC in different noise environments (SNR=5dB), (a) clean speech waveform, (b) VAD results by hands, (c)~(f) VAD results by BLZC in White, Babble, Factory, and Volvo noise, respectively

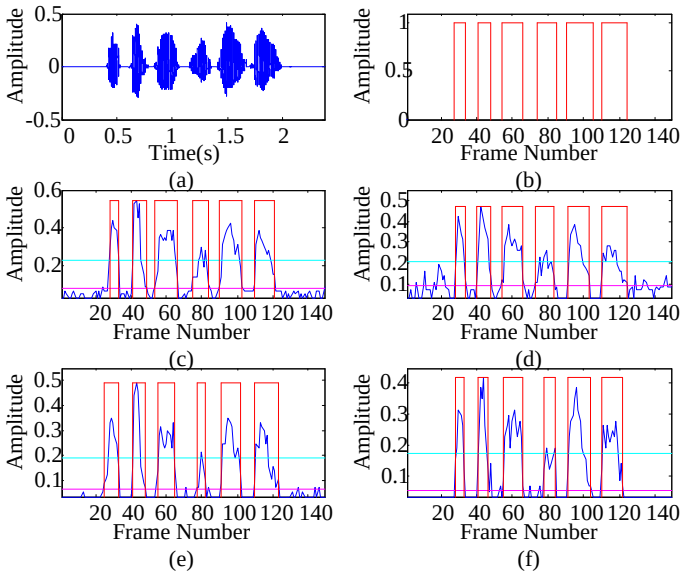


Fig. 11. VAD results by MLZC in different noise environments (SNR=5dB), (a) clean speech waveform, (b) VAD results by hands, (c)~(f) VAD results by MLZC in White, Babble, Factory, and Volvo noise, respectively

as a small amount of voice truncated error in the Factory and Volvo noise environments and can accurately detect the speech endpoints in other cases.

Due to truncation error and extended error which exist in the VAD, the experimental analysis is based on the Weighted Average Error measurement (WA) and its definition is [27, 32]:

$$\mathrm{WA}=\frac{K_C \cdot \mathrm{CLP} + K_W \cdot \mathrm{WDN}}{fNum}. \tag{20}$$

where, CLP stands for truncated error frame, WDN stands for extended error frame, and *fNum* means the total frame number of the sampled signal. K_C and K_W are weighting coefficients. According to people's subjective feelings, extended signals are more acceptable than truncated [32], so K_C=1.4, K_W=0.6. The WA results in different noise and SNR environments are shown in Table 1. It can be found that the WA values of MLZC in different situations are lower than BLZC, which demonstrates that MLZC has a better VAD performance than BLZC, and it also can be found MLZC in the Volvo noise has a lower WA, which indicates MLZC would have a good application prospect in the vehicle environment.

Table 1. WA results by two methods in different noise and SNR environments (%)

Noise	VAD	0dB	5dB	10dB	15dB	20dB
White	BLZC	26.71	13.67	9.26	8.19	7.65
	MLZC	12.21	11.28	7.52	6.57	6.04
Babble	BLZC	29.13	27.78	25.63	24.69	22.55
	MLZC	22.14	17.32	15.83	14.09	8.85
Factory	BLZC	57.85	57.71	50.58	48.38	46.93
	MLZC	25.90	21.61	19.73	17.72	9.79
Volvo	BLZC	8.18	7.78	6.97	5.91	5.50
	MLZC	6.97	6.58	5.63	4.81	4.25
Average	BLZC	30.47	26.73	23.11	21.79	20.65
	MLZC	16.81	14.19	12.17	10.79	7.23

5. Conclusions

In this paper, we propose a new VAD method that is multi-valued coarse-graining Lempel-Ziv Complexity (MLZC), which use fuzzy *c*-Means clustering algorithm and Bayesian information criterion algorithm to estimate the thresholds of the MLZC characteristic, and dual-thresholds method for VAD. Experimental results show that at low SNR environments, MLZC method is superior to the binary coarse-graining Lempel-Ziv Complexity (BLZC) method, especially in the vehicle interior noise environments, where MLZC method shows better detection performance. Therefore, we can say that MLZC method has a good application prospect and can provide accurate VAD techniques for car navigation.

In summary, there are several advantages that can be seen in the proposed VAD method: 1) Compared with the binary coarse-graining method, the multi-valued coarse-graining method can better perform the characteristics of speech signals. 2) We propose the novel non-linear feature of MLZC for VAD which could capture underlying model differences of speech and noise. 3) We use fuzzy *c*-Means clustering algorithm and Bayesian information criterion algorithm to estimate the thresholds, which more robust and heuristic-rules-free than previous thresholds estimation algorithms. In future work, we will apply the proposed VAD method to the speech recognition and speech applications in the car. Of course, it needs further verification.

Acknowledgements. This work is supported by Hunan Provincial Natural Science Foundation of P. R. China (Grant No.10JJ2046), and the Planned Science and Technology Key Project of Hunan Province, P. R. China (Grant No.2010GK2002).

References

1. Asano, F., Yamamoto, K., Hara, I., Ogata, J., Yoshimura, T., Motomura, Y.: Detection and separation of speech event using audio and video information fusion and its application to robust speech interface. EURASIP Journal on Applied Signal Processing 2004(11), 1727-1738 (2004)
2. Gilg, V., Beaugeant, C., Schonle, M., Andrassy, B.: Methodology for the design of a robust voice activity detector for speech enhancement. In: International Workshop on Acoustic Echo and Noise Control (IWAENC'2003). pp. 131-134. Kyoto, Japan (2003)
3. Tong, R., Ma, B., Lee, K.A., You, C. H., Kinnunen T., Sun, H.W.: Fusion of acoustic and tokenization features for speaker recognition. In: 5th International Symposium on Chinese Spoken Language Processing (ISCSLP'2006). pp. 566-577. Singapore (2006)
4. Zhang, L., Gao, Y. C., Bian, Z.Z., Lu, C.: Voice activity detection algorithm improvement in adaptive multi-rate speech coding of 3GPP. In: 1st International conference on Wireless Communications, Networking and Mobile Computing. vol. 2, pp. 1257-1260. Wuhan, China (2005)
5. Marko, T., Rosa, G.H., Pasi, F.: Automatic voice activity detection in different speech applications. In: Proceedings of the 1st international conference on Forensic applications and techniques in telecommunications, information, and multimedia and workshop. p. 12. Adelaide, Australia (2008)
6. Rabiner, L.R., Sambur, M. R.: Voice-unvoiced-silence detection using the Itakura LPC distance measure. In: Proceeding international Commence Acoustic. Speech, Signal Processing (ICASSP'1977). pp, 323-326. Hartford, Connecticut, USA (1977)
7. Haigh, J.A. Mason, J. S.: Robust voice activity detection using cpectral features. In: IEEE Region 10 Conference on Proceedings, Computer, Communication, Control and Power Engineering (TENCON'1993). pp. 321-324. Beijing, China (1993)
8. Junqua, J. C., Mark, B., Reaves, B.: A robust algorithm for word boundary detection in the presence of noise. IEEE Trans on speech and Audio Processing 2(3), 406-412 (1994)
9. Beritelli, F., Casale, S., Ruggeri, G., Serrano, S.: Performances evaluation and comparison of G.729/AMR/fuzzy voice activity detectors. IEEE Signal Processing Letters 9(3), 85-88 (2002)
10. Shen, J. L., Hung, J.W., Lee, L.S.: Robust entropy-based endpoint detection for speech recognition in noisy environments. In: International Conference on Spoken Language Processing (ICSLP'1998). pp. 232-235. Sydney, Australia (1998)
11. Huang, L.S., Yang, C.H.: A novel approach to robust speech endpoint detection in car environments. In: International Conference on Acoustics, Speech and Signal Processing (ICASSP'2000). vol. 3, pp. 1751-1754. Istanbul, Turkey (2000)

12. Zhao, H., Zhao, L.X., Zhao, K., Wang, G.J.: Voice activity detection based on distance entropy in noisy environment. In: Fifth International Joint Conference on INC, IMS and IDC. pp. 1364-1367. Seoul, Korea (2009)
13. Teager, H., Teager, S.: Evidence for nonlinear sound production mechanisms in the vocal tract. In: Proceeding of NATO ASI on Speech Production and Speech Modeling. pp. 241-261. Boston, USA (1990)
14. Christoph, B., Bemd, P.: Permutation entropy-a natural complexity measure for time series. PhysRev Lett, vol. 88, p. 174102 (2002)
15. Lei, X.G. Zeng, Y.C., Li, L.: Noisy speech endpoint detection based on approximate entropy. Technical Acoustics 26(2), 121-125 (2007)
16. Fan, Y.L., Wu, C.Y., Li, Y.: Application of C$_{0}$ complexity measure in detecting speech. Chinese Journal of Sensors and Actuators 19(3), 750-753 (2006)
17. Huang, H.Y., Lin F.H.: A speech feature extraction method using complexity measure for voice activity detection in WGN. Speech Communication 51(9), 714-723 (2009)
18. Nikhil, R.P., James, C.B.: On cluster validity for the fuzzy \emph{c}-means model. IEEE Transactions on Fuzzy Systems 3(3), 370-379 (1995)
19. Tian, Y., Wu, J., Wang, Z.Y.: Fuzzy clustering and Bayesian information criterion based threshold estimation for robust voice activity detection. In: International Conference on Acoustics, Speech and Signal Processing (ICASSP'2003). vol. 1, pp. 444-447 (2003)
20. Chen, S. S., Gopalakrishnan, P.S.: Clustering via the Bayesian information criterion with applications in voice recognition. In: International Conference on Acoustics, Speech and Signal Processing (ICASSP'1998). vol. 1, pp. 645-648 (1998)
21. Lempel, A., Ziv, J.: On the complexity of finite sequences. IEEE Transactions on information theory 22(1), 75-81 (1976)
22. Zhang, D.Z.: Research on the correlation between the mutual information and Lempel-Ziv complexity of nonlinear time series. Acta Physica Sinica 56(6), 3152-3157 (2007)
23. Zhang, X.S., Roy, R.J.: EEG complexity as a measure of depth of anesthesia for patients. IEEE Transactions on Biomedical Engineering 48(12), 1424-1433 (2001)
24. Liu, F.T., He, G.G.: Complexity measure for macroscopical transportation system using Lempel-Ziv algorithm. Journal of Harbin Institute of Technology 40(12), 2058-2061 (2008)
25. Schuster, H.G., Leiserson, C.E., Rivest, R.L.: Deterministic chaos: An Introduction. pp. 7-8. Wiley-VCH Verlag GmbH \& Co. KGaA, Weinheim, Germany (2005)
26. Rabiner, L., Juang, B.H.: Fundamentals of Speech Recognition. Prentice Hall, New Jersey, USA (1999)
27. Zhang, S.Y., Guo, Y., Zhang, Q.: Robust voice activity detection feature design based on spectral kurtosis. In: 1st International Workshop on Education Technology and Computer Science (ETCS'2009). pp. 269-272. Wuhan, China (2009)
28. John, S.G., Lori, F.L., William, M.F., Jonathan, G.F., David, S.P., Nancy, L.Da., Victor, Z.: DARPA TIMIT Acoustic-Phonetic Speech Corpus. Linguistic Data Consortium, Philadelphia (1993). [Online]. Available: http://www.ldc.upenn.edu /Catalog/CatalogEntry.jsp?catalogId=LDC93S1 (current Nov. 2010)

29. NOISEX-92 Database. [Online]. Available: http://spib.rice.edu/spib/select_noise.html (current Nov. 2010)
30. Voice Activity Detector (VAD) for adaptive multi-rate (AMR) speech traffic channels, ETSI, ETS1 EN 301 708 Recommendation (1999). [Online]. Available: http://www.3gpp.org/ftp/specs/html-info/0694.htm (current Nov. 2010)
31. A Silence Compression Scheme for G.729 Optimized for Terminals Conforming to Recommendation V.70, ITU, ITU-T Rec. G.729-Annex B (1996). [Online]. Available: http://eu.sabotage.org/www/ITU/G/G0729nbe1.pdf (current Nov. 2010)
32. Xu, W., Ding, Q., Wang, B. X.: A speech endpoint detector based on eigenspace energy entropy. Journal of China institute of communications 24(11), 125-132, (2003)

Huan Zhao is a professor at the School of Information Science and Engineering, Hunan University. She obtained her B.Sc. degree and M.S. degree in Computer Application Technology at Hunan University in 1989 and 2004, respectively, and completed her Ph.D. in Computer Science and Technology at the same school in 2010. Her current research interests include speech information processing, embedded system design and embedded speech recognition. She served as visiting scholar at the University of Califomia-San Diego (UCSD), USA during the period of March 2008 to September 2008. The visiting scholarship was appointed and sponsored by the China Scholarship Council (CSC). Prof. Zhao is a Senior Member of China Computer Federation, Governing of Hunan Computer Society, China and China Education Ministry Steering Committee Member of Computer Education on Arts. She has published more than 40 papers and 6 books.

Gangjin Wang received his B.Sc. degree in Mathematics and Applied Mathematics at the School of Mathematics and Information Science, Henan Polytechnic University, P. R. China in 2008. Currently, he is a M.S. candidate of Hunan University, P. R. China. His current research interests include speech information processing and voice activity detection.

Cheng Xu is a professor at the School of Information Science and Engineering, Hunan University. He received B.Sc. degree and M.S degree from Hunan University in P. R. China in 1983 and 1986 respectively. He received Ph.D. degree in Mechanical Manufacture and Automation from Wuhan University of Technology in P. R. China in 2006. His current research interests include embedded system, network multimedia applications and control theory.

Fei Yu was born in Ningxiang, P. R. China, on February 06, 1973. Before Studying in Peoples' Friendship University of Russia, Russia, he joined and worked in Hunan University, Zhejiang University, Hunan Agricultural University, P. R. China. He has wide research interests, mainly information technology. In these areas he has published above 50 papers in journals or

conference proceedings and a book has published by Science Press, China (Fei Yu, Miaoliang Zhu, Cheng Xu, et al. Computer Network Security, 2003). Above 30 papers are indexed by SCI, EI. He has won various awards in the past. He served as many workshop chair, advisory committee or program committee member of various international ACM/IEEE conferences, and chaired a number of international conferences such as IITA'07, ISIP'08, ISECS'08, ISIP'09, ISECS'09 and ISISE'08. He have taken as a guest researcher in State Key Laboratory of Information Security, Graduate School of Chinese Academy of Sciences, Guangdong Province Key Lab of Electronic Commerce Market Application Technology, Jiangsu Provincial Key Lab of Image Processing and Jiangsu Provincial Key Laboratory of Computer Information Processing Technology.

Received: September 06, 2010; Accepted: January 19, 2011.

DOI: 10.2298/CSIS101201033X

Worst Case Performance Bounds for Multimedia Flows in QoS-enhanced TNPOSS Network

Ke Xiong[1,2], Yu Zhang[1], Shenghui Wang[1], Zhifei Zhang[1], and Zhengding Qiu[1]

[1.] School of Computer Science & Information Technology, Beijing Jiaotong University, Beijing 100044, P.R. China
xiongke.bjtu.iis@gmail.com
[2.] Department of Electronic Engineering, Tsinghua University, Beijing, 100084, P.R. China
kxiong@tsinghua.edu.cn

Abstract. Network performance bounds, including the maximal end-to-end (E2E) delay, the maximal jitter and the maximal buffer backlog amount, are very important for network QoS control, buffer management and network optimization. QoS-enhanced To Next-hop Port Sequence Switch (QTNPOSS) is a recently proposed transmission scheme to achieve scalable fast forwarding for multimedia applications. However, the existing E2E delay bound of QTNPOSS network is not tight. To this end, this paper presents a lower E2E delay bound for QTNPOSS networks by using the network calculus theory, where the inherent properties (e.g. packet length and peak rate) of the flow are taken into account. Besides, the buffer size bound and the jitter bound of QTNPOSS network are also presented. Moreover, by extensive numerical experiments, we discuss the influences of the Long Range Dependence (LDR) traffic property and the Weighted Fair Queuing (WFQ) weight on the proposed network performance bounds. The results show that the WFQ weight influences the bounds more greatly than the LRD property.

Keywords: QTNPOSS network, performance bound, network calculus, fractal leak bucket, WFQ;

1. Introduction

Recently, network technology has been exploited so rapidly that more and more multimedia applications are expected to be delivered over packet networks. Such applications often have relative strict QoS requirements on network metrics, such as end to end (E2E) delay, jitter and packet loss probability. To cater for these transmission requirements, TNPOSS forwarding approach was proposed in [1], which adopts connection-oriented forwarding mechanism and works like explicit routing. Since TNPOSS network can perform scalable fast forwarding to achieve lower E2E delay, it is

capable of improving the QoS ability of packet networks. However, no extra QoS tools were designed for TNPOSS network to further improve its QoS performances. To enhance the QoS ability of TNPOSS network, two important components, i.e., the traffic shaping module and the queue scheduling block, were introduced into the original TNPOSS model was proposed. Furthermore, the worst case E2E delay bound of QTNPOSSwas give in [2].

However, the bound proposed in [2] is not tight, because the self-constraints including the maximum packet length and peak rate, of a flow were neglected. Since in network performance metrics, E2E delay is one of the most important targets of QoS provision and the E2E bound plays a very important role in network congestion control, buffer-size adjustment and scheduling optimization, a relative tighter E2E delay bound of QTNPOSS network is very worth being investigated. To this end, this paper shall focus on pursuit of a tighter E2E delay bound for QTNPOSS network. Moreover, as far as we known, no work has been done on the other QoS bounds for QTNPOSS network, such as the jitter bound and the maximal buffer length bound, so we shall also investigate these bounds in this paper.

The analysis tool we use is the Network Calculus theory [3], which is an very effective mathematical tool on analyzing network performances quantitatively in the worst case. Two key concepts within Network Calculus referred to here are the arrival curve and service curve, where the arrival curve is used to characterize the traffic feature of an arriving flow and the service curve is used to characterize process ability of a given network node.

The main contributions and novelties of this paper are: 1) giving a new arrive curve for a multimedia flow by considering its peak rate and maximal packet length; 2) modeling the E2E delay of QTNPOSS networks; 3) presenting a tighter E2E delay bound for QTNPOSS networks; 4) giving the maximal buffer length for a single node QTNPOSS network in the condition of no packet loss; 5) presenting the jitter bound for QTNPOSS network; 5) analyzing the parameters' influences on the QoS bounds of the QTNPOSS network by numerical experiments.

The rest of this paper is organized as follows: Section 2 will introduce the original TNPOSS network, the traffic shaping models, the WFQ scheduler and then the model of QTNPOSS network. In Section 3, we will introduce relate concepts of network calculus and then analyze the QoS bound of QTNPOSS network. In Section 4, extensive numerical experiments are performed to show the proposed E2E delay bound is tighter than existing one. Moreover, the parameters' influences on the QoS bounds are discussed in this section. Finally, Section 5 will summarize the work of this paper with some concluding remarks.

2. QTNPOSS Network

Since QTNPOSS network is the improved version of TNPOSS network, here

we shall introduce the original TNPOSS network at first.

2.1. TNPOSS network

In TNPOSS Network, a set of binary codes are used to identify the ports of a router. For example, consider the network shown in Figure 1, S and D are terminal devices. a, b, c, d and e represent the routing devices. The solid lines between any two nodes are the communication links. The binary code near each node is the port code of the corresponding link interface. Suppose $S \to a \to b \to d \to c \to e \to D$ is the selected path for delivering the packets from S to D, the path can be represented by the output port sequence 10 11 11 100 01 which actually consists of the ID code of the output link at each hop on the path.

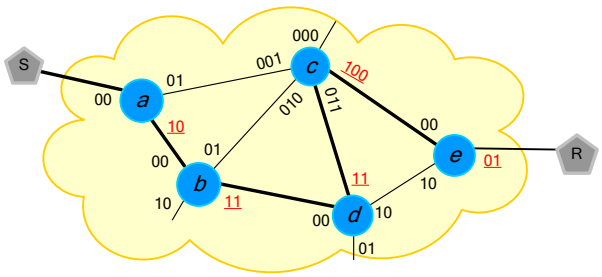


Fig. 1. TNPOSS network model

When a sender wants to send data to its destination, it will firstly initiate a request to setup one or more output-port code sequence paths. After this, the sender can use the output port sequence to do explicit routing. In TNPOSS network, since just during the path setup stage, routing tables need to be visited, while during the data transmission stage, each packet is only forwarded according to the output port code in the packet's header, the cost of routing lookup is greatly reduced and TNPOSS transmission can achieve fast switching. Moreover, TNPOSS is able to work on the basis of any routing protocol. If the routing protocol provides QoS routing, TNPOSS will transmit the packets on QoS-supported path. Since TNPOSS is able to deliver the packets of a flow on a pre-specified connection-based path, it has better QoS ability than today's IP networks. The detailed working process of TNPOSS network can be found in [1] and [9].

Although TNPOSS is suitable for transmitting multimedia flows due to its special working process and its explicit routing nature, it has no QoS tools to meet the strict QoS requirements of multimedia applications. Thus, authors in [2] proposed a QoS-enhanced version of TNPOSS network, i.e., QTNPOSS network. In QTNPOSS network, Fractal Leaky Bucket (FLB) model was

introduced to shape the arriving traffic and the Weighted Fair Queuing (WFQ) model was introduced to schedule the packet before output. Thus in next two subsections, we shall introduce the FLB model and WFQ model.

2.2. Fractal Leaky Bucket (FLB)

To enhance network QoS ability, traffic is required to be policed in order to guarantee that the sender does not send more than specified by the contract of a connection into the network. Policing devices inside network often do buffering, which are called shapers. One of the simple and effective shaper models is token bucket (TB) [4]. TB regulates the traffic by a linear function of a time interval τ. If we denote the traffic the sender transmits over the time interval τ with $A(\tau)$, the traffic is said to be regulated by TB, if there exists a pair (ρ, b) such that:

$$A(\tau) \le \rho\tau + b \text{ for any } \tau > 0, \quad (1)$$

where ρ represents the long-term average rate of the traffic, which is also the output rate of the TB, and b represents the maximum burst allowed to be sent into the network in any time interval, which characterizes the buffer size of the TB. Although TB has good ability to describe the characteristic of the linear bounded arrival processor, it can not describe the traffic of internet very well. The reason is that, most internet flows often have very high and stochastic burst rate and have Self-similar (SS) and Long Range Dependence (LRD) properties [5][6], but TB cannot characterize such properties of internet traffic. As for the LRD property, authors in [5][6] stated that it may bring down network performances, including increasing the E2E delay, the buffer size and the packet loss probability, etc. Thus, it is necessary to select a proper traffic shaper for TNPOSS network to support multimedia traffic well. Authors in [4] proposed the FLB model to regulate the LRD traffic instead of TB model, and the numerical results showed that FLB outperforms TB. Now, let give the description of FLB. If we denote the traffic the sender transmits over the time interval τ with $A^*(\tau)$, the traffic is said to be regulated by FLB, if there exists a pair (ρ^*, b^*) such that:

$$A^*(\tau) \le \rho^*\tau + b^* \text{ for any } \tau > 0, \quad (2)$$

where

$$\rho^* = \rho + \sigma(1-H)\sqrt{2\gamma\left(\frac{H}{1-H}\right)^{H-1}}, \quad (3)$$

and

$$b^* = \sigma(1-H)\sqrt{2\gamma\left(\frac{H}{1-H}\right)^{H}}, \quad (4)$$

where γ is a positive constant, whose value is usually assumed to be 6 [7]. σ is the standard variation of $A(\tau)$, and H is the self-similar parameter, which is in fact the burst parameter.

2.3. Weighted Fair Queuing (WFQ)

In this subsection, we will introduce the WFQ model, which is also a key element in QTNPOSS network. As is known, scheduling is one of the most important mechanisms to provide QoS guarantee in packet networks. One of the most notable scheduling models is the Generalized Processing Sharing (GPS) model [3][8]. GPS can control the sharing of one link among packets of different classes, but it is only method with idea assumptions and is not implementable. To approximate GPS, WFQ is considered as the most effective one, which does not have the assumption of infinitesimal packet size. So, WFQ scheduler was introduced into QTNOSS network. Now, let us give the mathematical description of WFQ. Suppose a WFQ scheduler serves N flows and each flow is specified by a positive weight w_i. $g_i(\tau)$ denotes the amount of served traffic of flow i in the time interval τ. If R is the service rate of the network node, then

$$g_i(\tau) = \frac{w_i}{\sum_{1 \le k \le N} w_k} * R * \tau \tag{5}$$

2.4. Node Model of QTNPOSS Network

From the description in Section 2.1, it can be seen that TNPOSS network performs QoS provisioning just via explicit routing mechanism and the output-port code based fast forwarding. No additional tools are provided to enhance the QoS ability of TNPOSS network. Thus, FLB shapers and WFQ schedules were added into the nodes of QTNPOSS. Figure 2 gives inner structure of the node of QTNPOSS Network. When a packet of a flow arrives at a node of QTNPOSS network, it is shaped by the FLB shaper, and then is scheduled by the WFQ scheduler according to the weight of the flow it belongs to. More detailed information about QTNPOSS network can be seen in [2] and [9].

Suppose the service rate of a QTNPOSS node is R, in terms of Eqn. (5), the service rate for flow i can be written as

$$V_i = \frac{Rw_i}{\sum_{i=1}^{N} w_k} \tag{6}$$

Moreover, according to [10], the maximum delay of WFQ model is

$$T_{wfq} = \frac{l_i}{V_i} + \frac{L_{\max}}{R}, \tag{7}$$

where l_i is the maximal packet length of flow i and $L_{\max} = \max_{1 \le j \le N}(l_j)$ is the maximal packet length of all flows in the node.

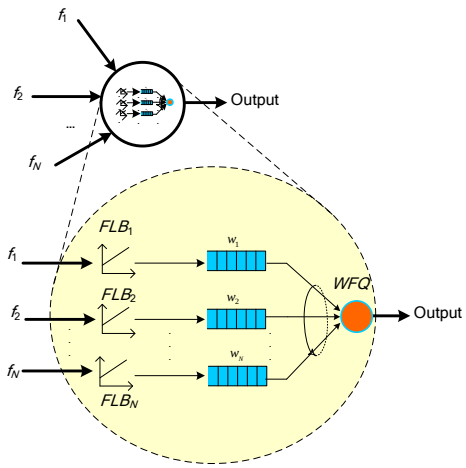


Fig. 2. QTNPOSS network model

3. Performance Bounds for QTNPOSS

In this section, we will analyze the QoS bounds for QTNPOSS network. The mathematical tool we use is the network calculus theory.

3.1. Network Calculus

Network calculus [3] is a collection of results based on Min-Plus algebra, which can be applied to analyze deterministic queuing systems in communication networks. Moreover, it is also a set of recent developments which provide a deep insight into flow problems encountered in networking, and is used with envelope bounded traffic models to provide a worst case analysis on network performance.

Note that network calculus is based on the idea that given a regulated flow of traffic into the network, one can quantify the characteristics of the flow as it travels from node to node through the network, which means that traffic flows are constrained by shapers and then delayed by the nodes' schedulers. In network calculus, the shapers are often modeled by *arrival curves* and the schedulers are modeled by latency *service curves*, so the key question is to formulate a correct arrive curve and a correct service curve for the analyzed system. Now we introduce some important concepts and conclusions of network calculus as follows at first.

Definition 1. (*WIF: wide-sense increasing function*). $f(x)$ is a function, for any $\forall s \le t$, if $f(s) \le f(t)$, f is a wide-sense increasing function.

The WIF is used to describe flow functions such as the $A(\tau)$ and $A^*(\tau)$ in this paper. Because if the packets of a flow arrive and departure bit by bit, for a duration of any period of time t, both the amount of the arrived flow traffic and that of the depastured traffic can be characterized by WIFs with respect to time t.

Definition 2. (*arrival curve*). Give a WIF α defined for a shaper, a flow f is constrained by α if and only if for all $s \le t$,

$$f(s) - f(t) \le \alpha(t-s). \tag{8}$$

The arrival curve actually defines an upper bound on the arrival rate of a flow to a particular node. The arrival curve of FLB is can be modeled by Eqn (2).

Definition 3. (*service curve*). If a system S has an input flow $f(t)$ and output flow $f_o(t)$, then S offers to the flow a service curve $\beta(t)$, if and only if for all $t \ge 0$,

$$f_o(t) \ge \inf_{s \le t}(f(s) + \beta(t-s)). \tag{9}$$

A service curve is a lower bound on the departure rate from a network node.

Definition 4. (*min-plus convolution*). Let f and g be two WIFs. The min-plus convolution of f and g is the function:

$$(f \otimes g)(t) = \begin{cases} \inf_{0 \le s \le t}[f(t-s) + g(s)], & t \ge 0 \\ 0, & t < 0 \end{cases} \tag{10}$$

Definition 5. (*min-plus deconvolution*). Let f and g be two WIFs. The min-plus deconvolution of f and g is the function:

$$(f \oslash g)(t) = \sup_{s \ge 0}[f(t+s) - g(s)] \tag{11}$$

Definition 6. (*virtual delay*). The virtual delay at time t is

$$d(t) = \inf\{\tau \ge 0 : \ f(t) \le f_o(t+\tau)\} \tag{12}$$

The virtual delay at time t is the delay that would be experienced by a bit arriving at time t if all bits received before it are served before it.

Definition 7. (*backlog*). If a system S has an input flow $f(t)$ and output flow $f_o(t)$, the backlog at time t is defined as

$$\phi(t) = f_o(t) - f(t) \tag{13}$$

The backlog is the amount of bits that are held inside the system; if the system is a single buffer, it is the queue length.

Definition 8. (*horizontal deviation*). Let f and g be two WIFs. The horizontal deviation $h(f,g)$ is defined as

$$h(f,g) = \sup_{t \ge 0}\{\inf\{\tau \ge 0; f(t) \le g(t+\tau)\}\} \tag{14}$$

Definition 9. (*vertical deviation*). Let f and g be two WIFs. The horizontal deviation $h(f,g)$ is defined as

$$v(f,g) = \sup_{t \ge 0}\{f(t) - g(t)\} \tag{15}$$

Definition 10. (*BDF: burst-delay function*). WIF $\delta_T(t)$ is called burst-delay function, if

$$\delta_T(t) = \begin{cases} 0, & t < T \\ +\infty, & t \geq T \end{cases} \tag{16}$$

Definition 11. (*RLF: rate-latency function*). WIF $\beta_{V,T}(t)$ is called rate-latency function, if

$$\beta_{V,T}(t) = V[t - T_{lat}]^+ = \begin{cases} V(t - T_{lat}) & t > 0 \\ 0 & t \leq 0 \end{cases}, \tag{17}$$

where T_{lat} is the latency delay and V is the service rate. The service curve of a GPS node can be represented by a RLF [3].

Property 1. (*service curve of concatenation nodes*). Assume a flow traverses systems S_1, S_2, …, and S_m in sequence. Suppose that S_i, $i \in [1, m]$, offers a service curve of β_i to the flow. Then the concatenation of the systems offers a service curve of $\beta_1 \otimes \beta_2 \otimes ... \otimes \beta_m$ to the flow.

Property 2. $d(t) \leq h(\alpha, \beta)$.

Property 2 shows that the virtual delay of a system is the horizontal deviation between its arrival curve α and service curve β.

Property 3. $\phi(t) \leq v(\alpha, \beta)$.

Property 3 shows that the backlog of a system is the vertical deviation between its arrival curve α and service curve β.

Property 4. $\delta_{T_1} \otimes \delta_{T_2} = \delta_{T_1 + T_2}$.

Property 5. $\beta_{V,T_1} \otimes \delta_{T_2} = \beta_{V,T_1+T_2}$.

Property 6. $\beta_{V_1,T_1} \otimes \beta_{V_2,T_2} = \beta_{\min\{V_1,V_2\},(T_1+T_2)}$.

Given three WIFs f, g and ζ, the following properties hold in network calculus.

Property 7. $f \otimes g = g \otimes f$.

Property 8. $f \otimes g \otimes \zeta = f \otimes (g \otimes \zeta)$.

Property 9. $f \oslash g \oslash \hbar = f \oslash (g \otimes \hbar)$.

Property 10. If g is sub-additive and $g(0) = 0$, then $g \oslash g = g$.

Property 11. $h(f, g) = \inf_{d \geq 0} \{d : (f \oslash g)(-d) \leq 0\}$.

More detailed information can be found in [6].

Recently, many researches begin to analyze network QoS performances using network calculus [4][11][12]. Authors in [4] discussed the E2E delay bound of the Expedited Flow defined in RFC 3246. Authors in [11] analyzed the E2E delay bound of the wireless sensor networks via the statistical network calculus and authors in [12] proposed the E2E delay bound for LRD flows under the shaping model of FLB, which obtained a similar conclusion to that presented in [2]. All obtained results show that network calculus has very strong ability for analyzing internet flows, which outperforms traditional mathematical tools, such as queuing theory, random process and probability theory, in terms of characterizing the queuing system of internet flows. Thus, in this paper, we shall go on using network calculus to derive QoS bounds for QTNPOSS network.

3.2. QoS Bounds of QTNPOSS Network

As mention in Section 2.4, each node of QTNPOSS network has a FLB shaper and a WFQ scheduler. Thus, the components of the E2E delay of QTNPOSS network can be divided into two classes. The first class is denoted by D_v consisting of the shaping delay and the scheduling delay, and the second class is denoted by D_s consisting of the link propagation delay and the node processing delay, referred to as *processing delay* hereafter. D_v has close relationship with the traffic of flow, so it is variable. In contrast, D_s is relatively stable, which can be seen as a constant. Thus, the E2E delay of a flow can be expressed as

$$D_{e2e} = D_s + D_v \tag{18}$$

Consequently, if we want to obtain D_{e2e}, the main question is to derive D_v.

Lemma 1. The arrive curve of a flow through a QTNPOSS network node is

$$\alpha^Q(t) = \min\{\rho^* t + b^*, pt + l\} \tag{19}$$

where ρ^* and b^* has the same meaning with those in Eqn. (2), and p represents the peak rate of the flow and l is the maximal packet length of the flow.

Proof. The arrive curve of a TB shaper can be expressed by $\rho t + b$ [3], so the arrive curve of a FLB shaper must be $\rho^* t + b^*$. Moreover, a flow through the shaper, apart from being regulated with the FLB, it also has some inherent features. That is to say, the flow also satisfies the constraints from its own features. Here, we find that the peak rate and the maximal packet length are two inherent features of flow. Thus, the flow must satisfies the lower value of $\rho^* t + b^*$ and $pt + l$, i.e., $\alpha^Q(t) = \min\{\rho^* t + b^*, pt + l\}$. □

Lemma 2. The service curve of a signal-node QTNPOSS network is $\beta^Q_{V,T_{wfq}}$.

Proof. As the service curve of all GPS-based scheduling model can be express by the RLF described in Eqn. (17). Moreover, the output rate and latency of QTNPOSS network node are determined by WFQ model. So Lemma 2 is proved. □

Now, let consider a simple case at first. Consider a QTNPOSS network contains only one node, the following Theorem 1 and Theorem 2 can be derived.

Theorem 1. (*single node QTNPOSS network E2E delay bound*). Suppose a LRD flow goes through a QTNPOSS network consisting of only one node, whose arrive curve and service curve are α^Q and β^Q, respectively. The maximal delay caused by this node satisfies that

$$D \le \frac{(V-p)^+ l - (p-V)^+ (\rho^* t_0 - V t_0 + b^*)}{(V-p)V} + T_{wfq} + D_s \tag{20}$$

where $t_0 = (b^* - l)/(p - \rho^*)$.

Proof. According to Eqn. (18) and Property 2, it can be inferred that

$D_{e2e} \le D_v + D_s \le h(\alpha^Q, \beta^Q_{V,T_{wfq}}) + D_s$. Moreover, if no packet is expected to be lost, $\rho^* \le V$ and $l \le b^*$. If we denote the arrive curve and service curve in a time-bit coordinate plane, the result can be seen in Figure 3.

When $V \ge p$,

$$h(\alpha^Q, \beta^Q_{V,T_{wfq}}) = \frac{l}{V} + T_{wfq}. \tag{21}$$

When $V < p$,

$$h(\alpha^Q, \beta^Q_{V,T_{wfq}}) = \frac{\rho^* t_0 + b^*}{V} + T_{wfq} - t_0 \tag{22}$$

If we combine Eqn. (21) with Eqn. (22) to form a unified expression, we can have that

$$h(\alpha^Q, \beta^Q_{V,T_{wfq}}) = \frac{(V-p)^+ l - (p-V)^+ (\rho * t_0 - V t_0 + b^*)}{(V-p)V} + T_{wfq}, \tag{23}$$

where $(x-y)^+$ means that if $x \ge y$, $(x-y)^+ = x-y$. Else, $(x-y)^+ = 0$. Thus, Theorem 1 is proved. □

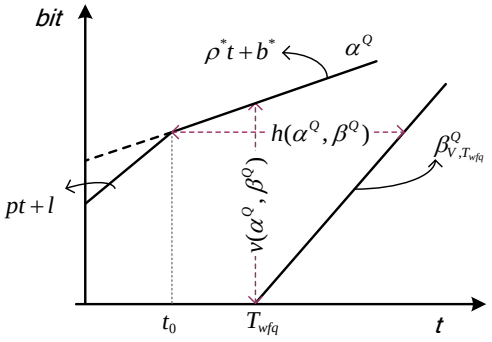


Fig. 3. Arrival curve and service curve of single node QTNPOSS network

Theorem 2. (*single node QTNPOSS network backlog*). Suppose a LRD flow goes through a QTNPOSS network consisting of only one node, whose arrive curve and service curve are α^Q and β^Q, respectively. The maximal backlog of the traffic within the network satisfies that

$$B \le \frac{(T_{wfq} - t_0)^+ (\rho^* T_{wfq} + b) - \frac{(t_0 - T_{wfq})^+ [(V-p)^+ (pT_{wfq} + l) - (p-V)^+ (VT_{wfq} + l + (p-V)t_0)]}{V-p}}{T_{wfq} - t_0} \tag{24}$$

Proof. According to the definition 7 and property 3, it is easily to be inferred that $B \le v(\alpha^Q, \beta^Q_{V,T_{wfq}})$. Moreover, from Figure 3, it can been seen that when $T_{wfq} \ge t_0$,

$$v(\alpha^Q, \beta^Q_{V,T_{wfq}}) = \rho^* T_{wfq} + b^* . \quad (25)$$

And when $T_{wfq} < t_0$, two cases should to be discussed. If $V \geq p$,

$$v(\alpha^Q, \beta^Q_{V,T_{wfq}}) = pT_{wfq} + l \quad (26)$$

else

$$v(\alpha^Q, \beta^Q_{V,T_{wfq}}) = pt_0 + l - V(t_0 - T_{wfq}) = (p-V)t_0 + l + VT_{wfq} . \quad (27)$$

If we express Eqn. (25), Eqn. (26) and Eqn. (27) with universal description, it can be derived that

$$v(\alpha^Q, \beta^Q_{V,T_{wfq}}) = \frac{(T_{wfq} - t_0)^+ (\rho^* T_{wfq} + b)}{T_{wfq} - t_0} + \frac{(t_0 - T_{wfq})^+}{t_0 - T_{wfq}} [\frac{(V-p)^+ (pT_{wfq} + l) - (p-V)^+ (VT_{wfq} + l)}{V - p} - (p-V)^+ t_0] .$$

Therefore, Theorem 2 is proved. □

Actually, the maximal backlog of a node also represents the minimal buffer length of a node, with which the single-node system can ensure no packet loss. Now we consider a more complex case, in which a QTNPOSS system is composed of many nodes rather than one. In order to keep the QoS ability of a multi-node QTNPOSS system, the concept of *greedy shaper* [13] is introduced. The greedy shaper is able to keep the packets of a flow away from loss and, at the same time, it can achieve the maximum output allowed by the shaper curve.

Lemma 3. Let S_Q be a QTNPOSS system characterized by the arrive curve α^Q and the service curve β^Q, and S_{QG} be a new system constructed by adding a greedy FLB shaper with a arrive curve of *αgs* into *SQ*, where the FLB shaper is placed between the shaper α^Q and the scheduler β^Q. Then, the E2E delay of a flow f traversing S_Q is equal to the E2E delay of f traversing S_{QG}. In other words, a greedy shaper does not increase the E2E delay of the system S_Q.

Proof. According to Property 2, the maximal E2E delay $D_{e2eS_{QG}}$ for f traversing the system S_Q is $h(\alpha^Q, \beta^Q)$. Meanwhile, it can be inferred that the maximal E2E delay $D_{e2eS_{QG}}$ for f traversing the system S_{QG} is $h(\alpha^Q, \alpha^{sg} \otimes \beta^Q)$. Since α^{sg} is greedy, from Eqn. (2) and Eqn. (20), it can be concluded that $\alpha^Q \leq \alpha^{sg}$. Thus, $\alpha^Q \oslash \alpha^{sg} \leq \alpha^Q \oslash \alpha^Q$. For a shaper, when $t = 0$, no traffic has come, so $\alpha^{sg}(0) = 0$. According to Property 10, $\alpha^{sg} \oslash \alpha^{sg} = \alpha^{sg}$. Furthermore, based on Property 9, $\alpha^Q \oslash (\alpha^{sg} \otimes \beta^Q) = \alpha^Q \oslash \alpha^{sg} \oslash \beta^Q = \alpha^Q \oslash \beta^Q$. Additionally, Property 11 indicates that $D_{e2eS_{QG}}$ can be written as

$$\begin{aligned} D_{e2eS_{QG}} &= h(\alpha^Q, \alpha^{sg} \otimes \beta^Q) = \inf_{d \geq 0} \{d: (\alpha^Q \oslash (\alpha^{sg} \otimes \beta^Q))(-d) \leq 0\} \\ &= \inf_{d \geq 0} \{d: (\alpha^Q \oslash \beta^Q)(-d) \leq 0\} \\ &= h(\alpha^Q, \beta^Q) = D_{e2eS_Q} \end{aligned} .$$

Hence, Lemma 3 is proved. □

Lemma 4. Let S_Q be a QTNPOSS system characterized by the arrive curve α^Q and the service curve *βQ*, and S_{QG} be a new system which is

constructed by adding a greedy FLB shaper with a arrive curve of α^{gs} in front of S_Q. Then, S_{QG} has the same arrive curve α^Q with S_Q.

Proof. Suppose the input flow is f and output flow if f_o. According to the definitionof arrive curve, it can be know that $f_o = f \otimes \alpha$. Moreover, Ref. [13] pointed that $f_o = f \otimes \alpha^{gs}$. Thus, it can be inferred that $f_o \le f \otimes \alpha^Q \otimes \alpha^{gs} = f \otimes \alpha^{gs} \otimes \alpha^Q = f_o \otimes \alpha^Q$. In other words, the arrive curve of S_{QG} also is α^Q. So, Lemma 4 is proved. □

Lemma 4 actually shows that a multi-node QTNPOSS network has the same arrival curve α^Q with the signal-node QTNPOSS system.

Theorem 3. Suppose a flow f traverse a multi-node QTNPOSS network, each node offers a service curve of $\beta^Q_{V_1,T_1}$, $(1 \le i \le H)$ where H is the number of the nodes on the path of flow f. The processing delay of node i is D_{s_i}. Then, the service curve of f provided by the whole multi-node QTNPOSS network is

$$\beta^{EQ} = \beta^Q_{\min\{V_1,V_2,...,V_H\},\sum_k^H (T_k + D_{s_k})} \tag{28}$$

Proof. Proof by mathematical induction. When $H=1$, $\beta_1^{EQ} = \beta^Q_{V_1,T_1} \otimes D_{s_1}$. According to Property 5, it can be obtained that $\beta_1^{EQ} = \beta^Q_{V_1,T_1+D_{s_1}}$. So, when $H=1$, Theorem 3 holds. Suppose $H = j-1$, where $j \ge 2$, Theorem 3 holds. Then,

$$\begin{aligned}
\beta_{j-1}^{EQ} &= \beta^Q_{\min\{V_1,V_2,...,V_{j-1}\},\sum_i^{j-1}(T_i + D_{s_i})} \\
&= \beta^Q_{\min\{V_1,V_2,...,V_{j-1}\},\sum_i^{k-1} T_i} \otimes \delta_{\sum_i^{j-1} D_{s_i}} \\
&= \beta^Q_{V_1,T_1} \otimes \beta^Q_{V_2,T_2} \otimes ... \otimes \beta^Q_{V_{j-1},T_{j-1}} \otimes D_{s_1} \otimes D_{s_2} \otimes \cdots \otimes D_{s_{i-1}}
\end{aligned}$$

When $H = j$, according to Property 1, it can can be induced that

$$\begin{aligned}
\beta_j^{EQ} &= \beta_{j-1}^{EQ} \otimes \beta^Q_{V_j,T_j} \otimes \delta_{D_{s_i}} \\
&= \beta^Q_{V_1,T_1} \otimes \beta^Q_{V_2,T_2} \otimes \cdots \otimes \beta^Q_{V_{j-1},T_{j-1}} \otimes \delta_{D_{s_1}} \otimes \delta_{D_{s_2}} \otimes \cdots \otimes \delta_{D_{s_{j-1}}} \otimes \beta^Q_{V_j,T_j} \otimes \delta_{D_{s_j}} \\
&= \beta^Q_{V_1,T_1} \otimes \beta^Q_{V_2,T_2} \otimes \cdots \otimes \beta^Q_{V_{j-1},T_{j-1}} \otimes \beta^Q_{V_j,T_j} \otimes \delta_{D_{s_1}} \otimes \delta_{D_{s_2}} \otimes \cdots \otimes \delta_{D_{s_{j-1}}} \otimes \delta_{D_{s_j}} \\
&= \beta^Q_{\min\{V_1,V_2,...,V_j\},\sum_i^j (T_i + D_{s_i})}
\end{aligned}$$

Therefore, when $H = j$, Theorem also 3 holds. Theorem 3 is proved. □

Theorem 4. Suppose a flow f traverse a multi-node QTNPOSS network, each node offers an arrival curve of α_i^Q and a service curve of $\beta^Q_{V_i,T_i}$, $(1 \le i \le H)$ where H is the number of the nodes on the path of flow f. The processing delay of node i is D_{s_i}. Then, maximum E2E delay of f traversing the multi-node QTNPOSS network must satisfy that

$$D_{e2e} \le \frac{(V_{\min} - p)^+ l - (p - V_{\min}\})^+ (\rho_1^* t_0 - V_{\min} t_0 + b_1^*)}{(V_{\min} - p) V_{\min}} + \sum_k^H (T_k + D_{s_k}), \tag{29}$$

where $V_{\min} = \min\{V_1, V_2, ..., V_H\}$.

Proof. Since flow f is shaped by the FLB when it comes into the network

and is constrained by its own property, the arrive curve of f provide by the first node must be $\alpha_1^Q = \min\{\rho_1^* t + b_1^*, pt + l\}$. Within the network, f will never be limited by (p,l), so $\alpha_k^Q = \rho_k^* t + b_k^*$, where $k \in [2,3,...,H]$. Moreover, according to Lemma 3 and Lemma 4, it can be known that α_k^Q will never increase the E2E delay of f, and f is always constrained by α_1^Q after being output by each node on the transmission path. Thus, the multi-node QTNPOSS network can be seen as a virtual system with arrival curve of α_1^Q and service curve of β^{EQ}. Then α_1^Q and β^{EQ} can be plotted on the time-bit coordinate plane, which is shown in Figure 4. From the formulation of β^{EQ} and Definition 11, it can be concluded that the T^E in Figure 4 must be $T^E = \sum_k^H (T_k + D_{sk})$. Since $D_{e2e} \le h(\alpha^Q, \beta^Q)$, on the basis of Figure 4, Theorem 4 can be proved easily. □

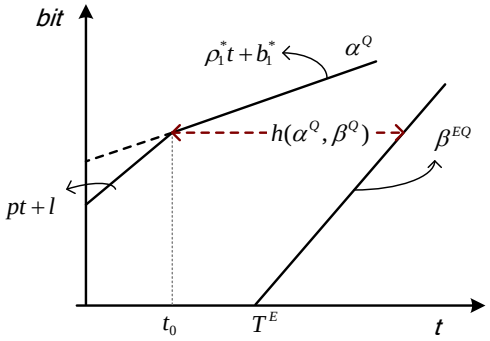


Fig. 4. Arrival curve and service curve of multi-node QTNPOSS network

Corollary 1. Theorem 1 is a special case of Theorem 4.

Proof. This corollary can be proved easily by assuming $H = 1$ in Eqn. (29). □

Corollary 2. $\sum_{i=1}^H D_{s_i} \le D_{e2e}$.

Proof. This corollary can be proved easily by the definition of D_{e2e}, where $\sum_{i=1}^H D_{si}$ is only one component of D_{e2e}. □

Theorem 5. Suppose a flow f traverse a multi-node QTNPOSS network, each node offers an arrival curve of α_i^Q and a service curve of $\beta_{Vi,Ti}^Q$, $(1 \le i \le H)$ where H is the number of the nodes on the path of flow f. The processing delay of node i is D_{s_i}. Then, maximum E2E delay jitter of f traversing the multi-node QTNPOSS network must satisfy that

$$J_{e2e} \le \frac{(V_{\min} - p)^+ l - (p - V_{\min}\})^+ (\rho_1^* t_0 - V_{\min} t_0 + b_1^*)}{(V_{\min} - p) V_{\min}} + \sum_k^H D_{s_k} , \qquad (30)$$

where $V_{\min} = \min\{V_1, V_2, ..., V_H\}$.

Proof. Since $D_{e2e} = \sum D_{v_i} + \sum D_{s_i}$, where only D_v can cause the jitter of delay, so $J_{e2e} \le D_{e2e} - \sum_k^H D_{sk}$. Thus, Theorem 5 is proved. □

4. Numerical Analysis

In this Section, we will analyze the parameters influences on the QoS bounds of QTNPOSS network. The parameters, including the self-similar parameter H, the normalized WFQ weight w and the number of nodes m are taken into consideration in our analysis. The other parameters and their values used in the numerical experiments of this section are listed in the Table 1, where ρ, p and γ are defined in Eqn. (2) previously. R is the service rate of the node in QTNPOSS network. Since we aim to analyze the QoS bounds in the worst case, we assume that all the nodes and links have the same processing delay respectively. Moreover, l and $L_{\max}$ are assumed to be the maximal value of the maximum transmission unit (MTU), which is about 1500 bytes.

Table 1. Parameters Setting in Analysis

Parameter Name	ρ	p	σ	γ	l	L_{max}	D_v	R
value	3×10^2	9×10^3	20	6	1500	1500	2	10
unit	kbps	kbps	kbit	–	Byte	Byte	ms	mbps

4.1. Comparison of E2E delay bounds

Based on the parameters mentioned above, we compare the proposed E2E elay bound shown in Eqn. (29) with the existing E2E delay bound in [2] within $w \in [0.1,1.0]$ and $H \in [0.5,0.95]$. The two bounds are expressed by two curved surfaces as shown in Figure 5. Obviously, it can be seen that the E2E delay bound proposed in this paper is tighter than the existing one. The reason is that, the properties of the flow are taken into account to be constraints.

Note that Ref. [3] has pointed out that the rationality of considering the inherent properties of a flow when one analyze the arrive curve for the flow. According to property 2 offered by network calculus, the worst case E2E delay can be derived. Thus, the proposed E2E delay bound is rational and has superiority over existing one. Moreover, as the jitter bound shown in Eqn. (30) can be seen as the result of the E2E delay bound subtracting a constant, i.e., D_s, the curved surface of the jitter bound will has the same tendency with the E2E delay bound shown in Figure 5.

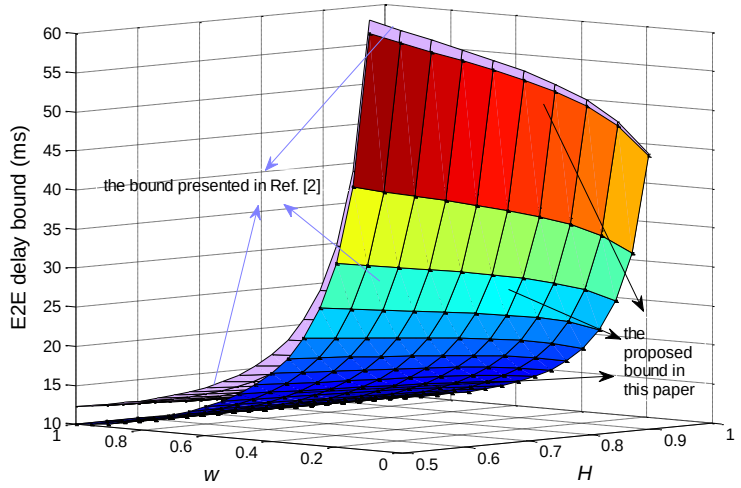


Fig. 5. Comparison of E2E delay bounds of multi-node QTNPOSS network

4.2. Analysis of proposed QoS bounds

In this subsection, we will discuss the influences of H and w on the QoS bounds.

QoS bounds of single node QTNPOSS network. Firstly, the QTPNOSS network is assumed to have only one node, which was introduced in Theorem 1. Figure 6 and Figure 7 show the influences of the H and the w on the buffer size bound of the single node QTNPOSS network, respectively. From Figure 6, it can be observed that for a given H the buffer size bound decrease with the increment of w. The reason is that the larger w is, the more bandwidth will be allocated for f. Thus, the data backlog will be reduced. Moreover, when H is relative large, the influence of w on the buffer size bound is small, and when H is relative small, the influence of w on the buffer size bound increases. From Figure 7, it can be seen that for a given w the buffer size bound also decrease with the increment of H. However, the influence of H on the buffer size bound seems much greater than that of w. Additionally, the curves marked with "1" in the two figures show the case of $T_{wfq} \geq t_0$, the curves marked with "2" show the case of $T_{wfq} < t_0$ and $V \geq p$, and the curves marked with "3" show the case of $T_{wfq} \geq t_0$ and $V < p$.

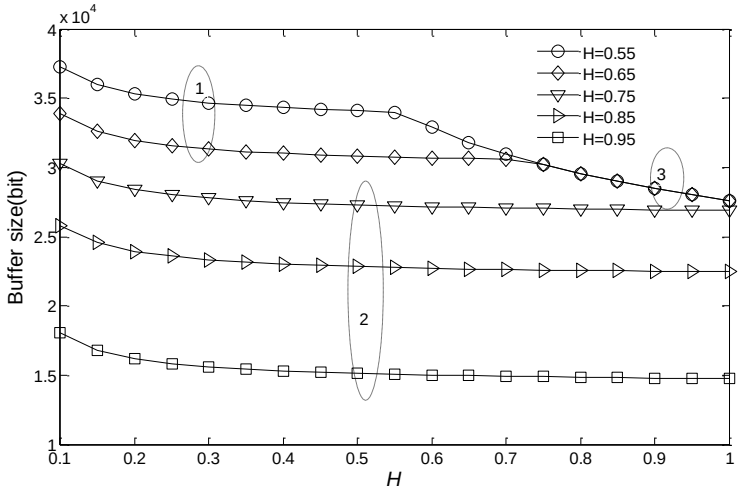


Fig. 6. Buffer size bound vs. w

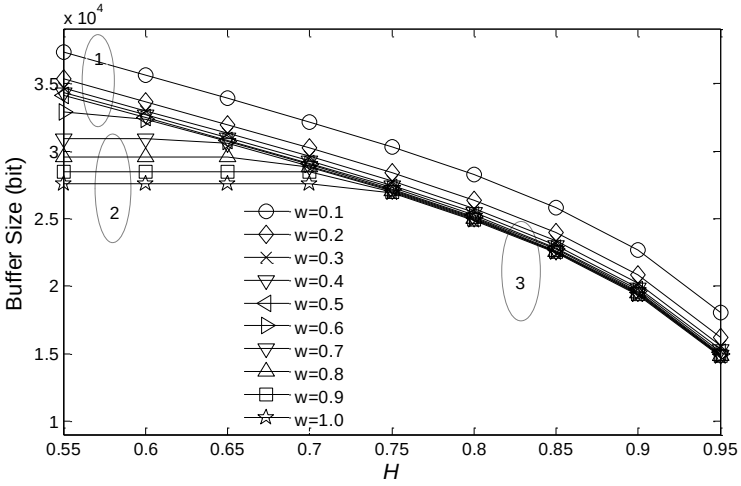


Fig. 7. Buffer size bound vs. H

Since the single-node QTNPOSS system is a special case of a multi-node QTNPOSS systems, so the impacts of H and w on the E2E delay of a single node QTNPOSS system are similar to those of H and w on the E2E delay of a multi-node QTNPOSS system, which can be seen in the next section.

QoS bounds of multi-node QTNPOSS network. In this subsection, a flow f is assumed to go through a path consisting of m nodes in the

QTNPOSS network. In the following experiments, m is assumed to be 15, because the measurement work in [14][15] show that the average number of hops between two communication nodes in Internet is about 15. Figure 8 and Figure 9 show the influences of H and w on the E2E delay bound of the QTNPOSS network, respectively. From Figure 8 and Figure 9, it can be observed that the E2E delay bound of Q TNPOSS network is influenced by both H and w. However, the influence of w is greater than that of H, especially when w is small. From all the experiments above, we can state that w influences the E2E delay bound more greatly than the other parameters do. The numerical results also indicate that by means of rising the degree of a flow's self-similar property or elevating the flow's WFQ weight, one can achieve acceptable low E2E delay.

5. Conclusions

This paper presented a lower E2E delay bound for QTNPOSS network by using network calculus. Besides, the buffer size bound, and the jitter bound of QTNPOSS network are also presented. To obtain these QoS bounds, the inherent properties (e.g. packet length and peak rate) of a flow were taken into account. We gave the arrive curve and service curve. Extensive numerical experiments show that both the long-range dependence property and the WFQ weight have influence on the E2E delay bound, and the WFQ weight has greater influence on the E2E delay bound than that of the long-range dependence property.

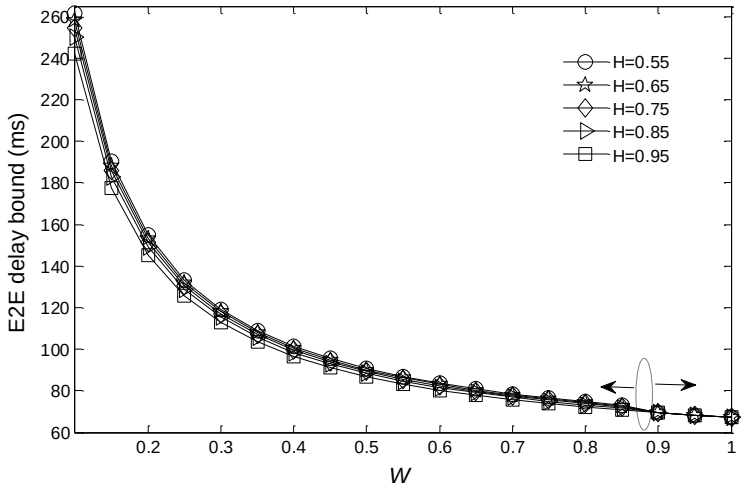


Fig. 8. E2E delay bound vs. w

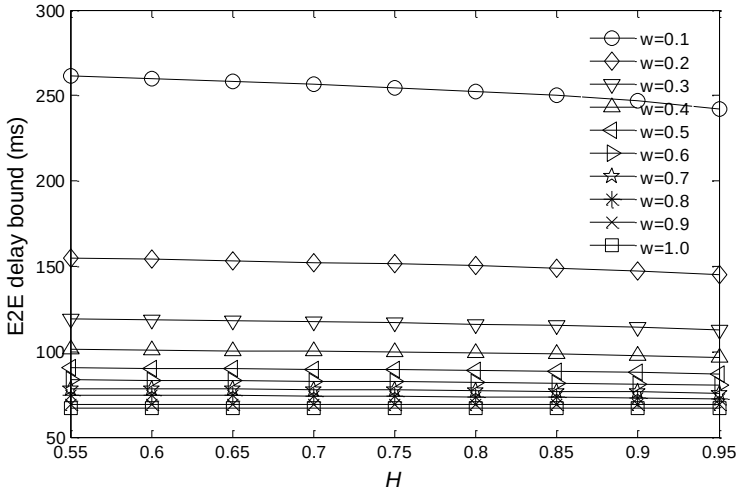


Fig. 9. E2E delay bound vs. H

Acknowledgment. The authors wish to give many thanks to Prof. Hong-ke Zhang, Prof. Ya-juan Qin, Prof. Yu-chun Guo and Dr. Gao, all of whom are with the School of Electronic and Information Engineering, Beijing Jiaotong University, China. This paper is supported by the China Postdoctoral Science Foundation No. 20100480329, partially by the National Basic Research Program of China (”973 program”) under contract No. 2007CB307101 and No. 2007CB307106, partially by the National High-Tech Research and Development Plan of China under Grant No. 2007AA01Z202, partially by the Program of Introducing Talents of Discipline to Universities (”111 Project”) under contract No. B08002, partially by the Cultivation Fund of the Key Scientific and Technical Innovation Project, Ministry of Education of China under contract No. 706005, and partially by the National Natural Science Funds under grant No. 60772043.

References

1. Xiong, K., Qiu, Z. D., Zhang, H. K., Li, C. S., Chao, H. C.: A scalable fast forwarding approach for IP networks. International Journal of Internet Protocol Technology, vol. 3, no. 2, pp. 119-127. (2008)
2. Xiong, K., Zhang, Y., Qiu, Z. D., SUN, D. M., Qin, Y. J.,: QoS-enhanced TNPOSS Network Model and its E2E Delay Bound for Multimedia Flows. Journal of Networks, vol. 4, no. 8, pp. 795-802. (2009)
3. Le Boudec, J. Y., and Thiran, P.: Network Calculus, Berlin: Springer Verlag, Version May 10. (2004)
4. Procissi, G., Garg, A., Gerla, M., and Sandadidi, M. Y.: Token bucket characterization of long-range dependent traffic. Computer Communications, vol. 25, no. 11-12, pp. 1009-1017. (2002)
5. Le Boudec, J. Y., Some Properties Of Variable Length Packet Shapers.

ACM/IEEE Transactions on Networking, vol. 10, no. 3, pp. 329-337. (2002)
6. Chakraborty, D., Ashir, A., Suganuma, T., Mansfield Keeni, G., Roy, T. K., Shiratori, N.: Self-similar and fractal nature of internet traffic," International Journal of Network Management, vol. 14, no. 2, pp. 119-129. (2004)
7. Zhang, L. M., Huang, D. Z., Chen, Z. G.: Models of bounds on end-to-end delay of long-range dependence traffic based on fractal leak bucket, Journal on Communications, vol. 29, no. 7, pp. 32-43. (2008)
8. Parekh A. K., and Gallager, R. G.: A generalized processor sharing approach to flow control in integrated service networks: The single node case. IEEE/ACM Transactions on Networking, vol. 1, no. 3 , pp. 344-357. (1993)
9. Xiong, K., Qiu Z. D., Guo Y. C., Zhang H. K., Chao, H. C.: Two LIC-Based Fast Forwarding Schemes for Explicit Routing with Scalability, Flexibility and Security. Journal of Internet Technology, vol. 12, no. 3, pp. 407-416. (2011)
10. Towela, N. J.: Impact of Traffic Aggregation on Network Capacity and Quality of Service. University of Kansas. http: //www.ittc.ukans.edu. (2002)
11. Qi, H. M., Chen, Z. Q., Zhang, L. M.: Towards end-to-end delay bounds on WMNs based on statistical network calculus. In: Proceedings of the 9th International Conference for Young Computer Scientists, pp. 493-497. (2008)
12. Zhang, L. M., Huang, D. Z., Chen, Z. G.: Models of bounds on end-to-end delay of long-range dependence traffic based on fractal leak bucket. Journal on Communications, vol. 29, no. 7, pp. 32-43. (2008)
13. Zhang, X. M., Chen, G. L., Gu, J.: A Traffic Shaping Framework Based on Network Calculus. Journal of Software, vol. 13, no. 12, pp. 2225-2230. (2002)
14. Ma, J. G., Xi, M. X., Lin,Y. M., Li,Y. P.: Chinese Internet routerlevel hop count measurement and analysis. Application Research on Computer, vol. 25, no. 27, pp. 2112-2114. (2008)
15. Begtasvi, F., Van Mieghem, P.: Measurements of the Hopcount in Internet. In: Proc. of Passive and Active Measurement (PAM 2001), Amsterdam, the Netherlands, April 23-24, pp. 183-190. (2001)

Ke Xiong received the B.S. and Ph.D. degrees in School of Computer and Information Technology, Beijing Jiaotong University, Beijing, R.P. China, in 2004 and 2010, respectively. He is currently a Post Doctor with the Department of Electronic Engineering, Tsinghua University, Beijing, China. His research interests are in the areas of next generation Internet, network QoS management, wireless relay network, network information theory, network coding, etc.

Yu Zhang received his B.S. degree in computer science from Beijing Jiaotong University (formerly knows as Northern Jiaotong University), Beijing, China, in June 2004. He is currently working toward the Ph.D. degree at the Institute of Information Science, School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China. His research interests include digital signal processing, multimedia communication & processing and compressed sensing.

Shenghui Wang received his Ph.D. degree in Digital Signal Processing from Beijing Jiaotong University in 2007. He is now a lecturer at Institute of Information Science, Beijing Jiaotong University. His main research interests include video coding & transmission, network traffic modeling and prediction, embedded technology, digital signal processing, etc.

Zhengding Qiu recieved B.S. degree in Communication Engineering in 1967, and M.S. degree in Digital Signal Processing in 1981 from Northern Jiaotong University, Beijing respectively. During 1967 to 1978, he was a communication engineer at LiuZhou Railway Bureau. Since 1981, he worked at Institute of Information Science, Northern Jiaotong University, where he became a lecturer in 1983, an associate professor in 1987 and a full professor in 1991. He was an oversea research scholar at Commputer Science Department of Pittsburgh University, USA, and joined in the program of parallel processing in signal processing. During 1999 to 2000, he was an visiting research fellow at Electronic Engineering Lab of Kent University, UK, and engaged in image prosessing and biometrics processing program. His reserach field includes digital signal processing, multimedia communication and processing and parallel processing. He is currently a fellowship of Chinese Institute of Communications, a senior member of Chinese Institute of Electronics and Railway respectively.

Received: February 01, 2010; Accepted: February 19, 2011.

DOI: 10.2298/CSIS101223034X

A Two-Tiered Reliable Application Layer Multicast

Xinchang Zhang[1], Meihong Yang[1], Guanggang Geng[2],
Wanming Luo[2], and Xingfeng Li[3]

[1] Shandong Key Laboratory of Computer Networks,
Shandong Computer Science Center,
Jinan, 250014, China
xinczhang@hotmail.com, yangmh@keylab.net
[2] Computer Network Information Center,
Chinese Academy of Sciences,
Beijing, 100190, China
{gengganggang, luowanming}@cnnic.cn
[3] The People's Bank of China,
Beijing, 100800, China
li_xingfeng@126.com

Abstract. This paper presents a two-tiered reliable application layer multicast (ALM) solution, called HRALM, to provide lossless ALM services. HRALM builds a domain-based multicast tree, and divides the members into two transport planes (i.e., Plane 1 and Plane 2) in terms of the tree. In any domain, the distance between each member and the domain header is below a given threshold, which improves the capability of topology-awareness. According to the loss detected by members in different planes, HRALM adopts different but correlated recovery solutions. In HRALM, a member duplicates and forwards the received recovery packet to each of its children if it has not received the data unit carried by the recovery packet before receiving the recovery packet, which can actively recover the loss at the downstream nodes. The simulation experiments show that HRALM has desirable transport and recovery performance.

Keywords: application layer multicast, negative acknowledgement, loss, recovery.

1. Introduction

In group applications (e.g., file distribution and multi-party game), multicast is the most efficient approach because it saves much bandwidth and greatly reduces the load of servers. Multicast functionality was originally implemented at the IP layer. IP multicast is an excellent approach to deliver multicast packets, without any unnecessary data duplication. However, IP multicast has some drawbacks that are the hurdles to its ubiquitous deployment. For examples, IP multicast depends on the support of multicast routers, and IP

multicast can make the entries of route forwarding tables increase rapidly. More problems of IP multicast can be seen in [1].

As an alternative of IP Multicast, application layer multicast implements the multicast functionality at application layer instead of IP layer. In ALM, network infrastructures need no additional modification, which addresses the problem of non-ubiquitous deployment of IP Multicast across wide-area. A major disadvantage of building ALM trees is that the members have no direct knowledge of the underlying topology, which brings some unavoidable performance penalties. In other words, ALM accelerates multicast deployment at the cost of acceptable performance penalties (such as additional traffic load and latency). Presently, ALM has been widely researched (see [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]).

In the delivery tree of IP multicast, branch nodes (other than the root) are multicast routers, and leaf nodes are group members. In contrast, no-leaf nodes in ALM trees are dynamic group members instead of multicast routers. Therefore the transport in ALM is obviously unreliable. In the following parts, reliable ALM refers to the ALM that can provide lossless service [16]. Providing end-to-end reliability through TCP is a choice of implementing the reliable ALM. However, it is difficult for this approach to obtain effective flow control in the viewpoint of group communication, because the end-to-end transmissions are asynchronous. For the same reason, the approach requires frequent data numbering and renumbering operations. Additionally, some end-to-end transmissions can be broken because group members can leave the group randomly [17]. LER [16] is one of few available studies on reliable ALM based on UDP. LER employs a lateral retransmission instead of a vertical retransmission from a host's ancestors. Since LER randomly divides hosts into several planes and independently builds an overlay tree in each plane, the capability of clustering nearby nodes is limited in some degree. Another limitation of LER is that it takes high measurement and computation overheads to select proper recovery neighbors [17].

This paper analyzes the transmission features of ALM delivery trees, and further proposes a two-tiered reliable application layer multicast solution HRALM. HRALM builds a domain-based multicast tree. A domain consists of three types of members, i.e., domain header (DH), domain agent (DA) and common host (CH). In each domain, the distance between each member and the domain header is below a given threshold. The distance-based domain can improve the capability of topology-awareness. HRALM divides the members into two transport planes (i.e., Plane 1 and Plane 2) in terms of the built delivery tree. Specifically, Plane 1 consists of all the domain headers, and Plane 2 consists of other members. HRALM employs different but correlated solutions to recover the loss detected by members in different planes. HRALM also uses an active recovery mechanism to improve the recovery performance. The members in the two planes play obviously different roles, and the approaches for recovering the loss at members in the two planes are also different. Therefore, HRALM can be considered as a two-tiered reliable ALM solution.

The rest of the paper is organized as follows. In Section 2, we introduce the related work. Section 3 presents an overview of HRALM architecture. HRALM′s tree building and loss recovery approaches are explained in Section 4 and 5, respectively. We evaluate HRALM performance by analyzing the simulation results in Section 6. Finally, we conclude our work in Section 7.

2. Related Work

ALM is a promising solution to provide the delivery service to group applications. Presently, ALM has been widely researched. In the delivery tree of IP multicast, branch nodes (other than the root) are multicast routers, and leaf nodes are group members. However, no-leaf nodes in ALM delivery trees are dynamic group members instead of multicast routers, which make the delivery of ALM unreliable. Therefore the reliable application layer multicast technology is an important research topic.

To improve the reliability, multiple-tree multicast approaches have been proposed, e.g., CoopNet [18], SplitStream [19], THAG [20] and NHAG [21]. Multiple-tree multicast constructs multiple paths between the root and each group member and delivers descriptions by using MDC [22] [23] to split the original streaming media into several descriptions. CoopNet proposes a centralized algorithm to facilitate deployment of multiple-multicast trees from different sources, and does not have explicit mechanisms to maximize bandwidth. SplitStream is a tree-based multicast algorithm based on structured overlay networks. THAG and NHAG can construct the node-disjoint multicast tree. Though the multiple-tree multicast approaches can improve the reliability of ALM, they are not reliable ALM solutions.

In ALMI [3], data distribution along the multicast tree occurs on a hop by hop fashion. Depending on the application, the data transfer between two adjacent members can be reliable or unreliable by deploying TCP or UDP, respectively. Yoid [8] also gives a similar TCP-based scheme for providing the reliable service. [17] explains some cases where TCP cannot provide good end-to-end reliability for group members in ALM environment, including: If a member leaves or fails, all the member's descendants need to reconnect to the remaining overlay and establish new TCP sessions from where they stopped; While it's easy to reconnect to the overlay, it's not guaranteed that the data flows can be restarted from where they are stopped; If the buffer of a member host has finite size, the packets needed by the newly established TCP session might not be in the buffer. [16] points out that (1) TCP-based reliable approach may not achieve high throughput due to TCP backoff mechanism, (2) the hosts at the leaves of the delivery tree may suffer from high delay, as a data segment has to be completely received before being forwarded downstream, and (3) it is not obvious to extend TCP in hop-by-hop, packet-by-packet manner for the reliable service.

LER [16] is one of few available studies on reliable ALM based on UDP. LER employs a lateral retransmission instead of a vertical retransmission from

a host's ancestors. LER randomly divides hosts into several planes and independently builds an overlay tree in each plane. In a plane, a host acts as the multicast tree root (i.e., the plane source). The original source sends data to all the plane sources, which then distribute data along their own trees. Each host selects some hosts in other planes as its recovery neighbors, which are sorted according to the estimated recovery latency. A limitation of LER is that it takes high measurement and computation overheads to select proper recovery neighbors [17]. Clustering nearby nodes is a promising approach for building the delivery tree with low end-to-end delay and network traffic. However, randomly dividing the hosts into some planes weakens the above advantage to some extent.

In contrast, providing reliable service based on IP multicast has been widely studied (see [24, 25, 26, 27]). The NACK-based recovery mechanism is widely adopted in the existing approaches for reliable IP multicast. In the NACK-based mechanism (e.g., NORM [25]), the receiver (i.e., the group member other than the root in the delivery tree) sends a NACK message to request the receiver (of the message) to retransmit the recovery packet. The NACK-based recovery mechanism reduces the retransmission delay to some extent. Because of the above intrinsic difference, it is unwise for reliable ALM to directly leverage some existing approaches that work well in reliable IP multicast.

3. Overview of HRALM Architecture

The design objective of HRALM is to provide reliable (i.e., lossless) ALM services to the group members. Specifically, HRALM builds an ALM tree to distribute the data and recovers the loss in the distribution procedure. HRALM uses our proposed TCM model (see [28]) to build the multicast tree and divides the group members into two transport planes in terms of the built tree. In the structure of the HRALM tree, there are many distance-based domains, and most of group members each belong to a domain. Each domain has a center, called domain header (DH). In any domain, the distance between each member and the domain header is within a given cluster threshold (denoted by λ). In HRALM, there exist some members, called foreign hosts (FHs), which do not belong to any domain. Except DH and FH, there also are two types of members, i.e., domain agent (DA) and common host (CH). Fig. 1 illustrates the four types of members:

- Domain header: DH is the center of the corresponding domain, i.e., the distance between each member in the domain and the header is not more than the given domain threshold λ. The child of a DH member might be (1) a CA in the same domain, or (2) a DH in a different domain, or (3) a FH which does not belong to any domain.

- Domain agent: DA is a child of the DH of the corresponding domain. DA may accept three types of members (i.e., CHs in the same domain, FHs, and DHs in different domain) as its children.
- Common host: The parent of each CH is a DA node, and only other CHs in the same domain and FHs can become the children of a CH node.
- Foreign host: A FH (denoted by *r*) does not belong to any domain, and the parent of *r* might be a DH, DA, or CH. Let *h* mean the first upstream DH node in *r*'s root path, then the distance between *r* and *h* is more than λ. Only other FHs can become the children of a FH.

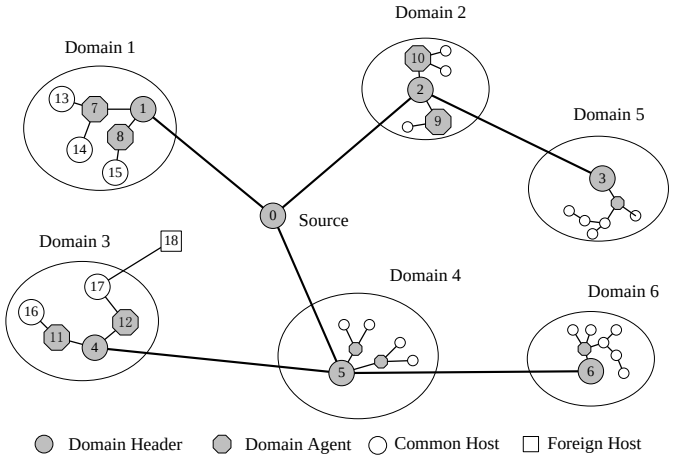


Fig. 1. Structure of the HRALM tree

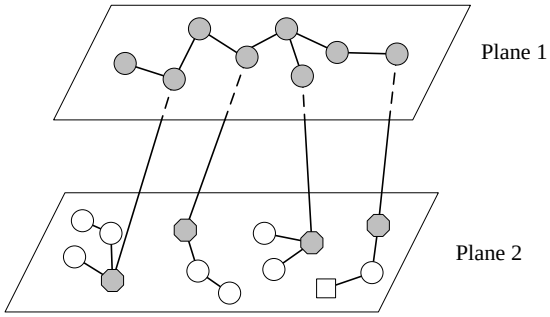


Fig. 2. The two-tiered transport planes of HRALM

According to the above description, we know that a domain consists of three types of members, i.e., DHs, DAs, and CHs. In the following parts, domain members represent the members (except the DH) in a certain domain. In HRALM, each member keeps the corresponding addresses and types of its children. If a child is *X* (DH, DA, CH, or FH) node, we call it *X*

child. Additionally, we say that a domain *D* is leaded by member *m* if *m* is the domain header of *D*.

The tree root (i.e., the data source) is a special DH member, and the domain leaded by the root only contains a single member (i.e., the DH). In HRALM, there are two types of transport planes, i.e., Plane 1 and Plane 2. All the DH members belong to Plane 1, while other members belong to Plane 2. Fig. 2 shows the structure of the above two-tiered transport planes.

In the application layer multicast, clustering nearby members can make the tree structure congruent to the network topology to some extent. Based on the above heuristic, HRALM adopts the above structure to cluster the members. Another important reason of using the above structure is to form the two-tiered transport planes in terms of the types of members.

In the reliable multicast, the confirmation entity is either packet or ADU, as IETF RFC 2887 [29] explains. HRALM uses ADU-level confirmation, which corresponds to ALM in nature. In HRALM, there are two types of negative acknowledgement packets, i.e., NACK1 (sent by members in Plane 1) and NACK2 (sent by members in Plane 2). Since the members in Plane 1 are usually in the top of the HRALM tree, HRALM attempts to recover the loss at members in Plane 1 through a quick and robust approach. The above quick and robust recovery ensures that the nodes in Plane 1 are relatively reliable. Based on these relatively reliable nodes, HRALM uses multi-round approach to recover the loss at members in Plane 2. Through the above hierarchical recovery solution, the recovery performance in HRALM is effectively improved.

HRALM also use an active recovery mechanism, i.e., any member duplicates and forwards the received recovery packet if it has not received the data unit carried by the packet before the arrival of the packet. In ALM, a loss at some tree node (denoted by *m*) must result in the same loss at the downstream nodes of node *m*. Therefore the above mechanism can actively recover the loss at the downstream nodes and reduce the recovery delay.

For providing lossless delivery service, the data source in HRALM buffers all the sent ADUs. However, the member (other than the root) buffers the latest received ADUs in terms of a fixed buffer size designed by the member.

4. Tree Construction of HRALM

In this section, we will introduce the TCM-based tree building approach used in HRALM. In the approach, the group information (e.g., the address of the root) is memorized by a Rendezvous Point (RP). When the newcomer wants to join the group, it first contacts the RP and gets the address of the data source. Then the newcomer joins the group in terms of the join algorithm shown in Fig. 3.

In the join algorithm, *candi* represents a candidate node which might become the parent of the newcomer. The join algorithm uses round trip time (rtt) as the distance metric. In this paper, we use $d(m,n)$ to denote the round

trip time from *n* to *m*. In the algorithm, JoinDomain(*m*,*n*,1) means host *n* joins the domain whose header is *m*, and JoinDomain(*m*,*n*,2) represents *n* become a downstream FH node of *m*. We will give more explanation on the above two procedures in next part. In terms of Line 3 in the join algorithm, *n* cannot become a domain member of the domain leaded by *s* if $d(s,n) \leq \lambda$. Line 4 in the algorithm is executed if (1) $d(m,n)>\lambda$ and $f(n) \geq 2$, or (2) $d(m,n) \leq \lambda$ and $m=s$. In HRALM, there are three types of join messages, i.e., JoinRequest (*t*,0), JoinRequest (*t*,1) and JoinRequest (*t*,2), where *t* denotes the receiver of the join messages. If a node accepts a join request from a member of type A, then the new child of the node is marked with symbol A.

Procedure Join (*s*,*n*)

// Newcomer *n* joins the group. *s* denotes the data source.

1: Initialize：*candi*←*s*; *S* is allocated for storing past *candi* nodes. All members have no any label by default. // *S* is a stack
2: Query *candi* to discover all its DH children. Measure the rtts from *candi* and its DH children to *n*. Let Ω mean the set of *candi* and its DH children.
3: Find the nearest node (*m*) among nodes in Ω. If $d(m,n) \leq \lambda$ and $m \neq s$, then JoinDomain(*m*,*n*,1); If $d(m,n)>\lambda$ and $f(n)<2$, then JoinDomain(*m*,*n*,2).
4: Find the nearest member (denoted by *m'*) among nodes (without the *full* labels) in Ω. If all nodes in Ω are marked with a *full* label, then pop the top element *p* of *S*, *candi*←*p*, go to Line 2.
5: If *m'* is not the current candidate node, push *candi* onto the stack *S*, *candi*←*m'*, go to Line 2.
6: Send a join request JoinRequest (*candi*,0) to *candi*. If the join request is refused, then mark *candi* with a *full* label and go to Line 4. Otherwise, *candi* becomes *n*'s parent node, and *n* mark itself with a DH label.

Fig. 3. Join algorithm

In this paper, we use $f(m)$ to denote the fanout of member *m,* i.e., the maximum number of children which host *m* is willing to accommodate in the multicast tree [2]. Similarly, we use $f'(m)$ to denote the remnant fanout of *m*. $f'(m)$ is equal to the number of existing children subtract $f(m)$. The fanout of member *m* is obtained by sending some detecting packet before *m* joins the group. Let $\sigma_1(k)$, $\sigma_2(k)$, $\sigma_3(k)$ and $\sigma_4(k)$ denote the number of DH children of *k*, the number of DA children of *k*, the number of CH children of *k* and the number of FH children of *k*, respectively. Then a DH (donated by *d*) responds to a join request sent by *t* in terms of the following cases:

(1) If $f'(d) \geq 2$, *d* accepts join request of any type.
(2) If $f'(d)=1$, $\sigma_2(d) \geq 1$ and *d* receives a JoinRequest(*d*,0) message, then *d* accepts *t* as its DH child.

(3) If $f'(d)=1$, $\sigma_1(d) \geq 1$, and d receives a JoinRequest(d,1) message, then d accepts t as its DA child.
(4) If $f'(d)=0$, $\sigma_4(d) \geq 1$, and d receives a JoinRequest(d,0) message, then d: accepts t as its DH child; randomly selects a FH child and tells it to rejoin the group starting from d.
(5) If $f'(d)=0$, $\sigma_4(d) \geq 1$, and d receives a JoinRequest(d,1) message, then d: accepts t as its DA child; randomly selects a FH child and tells it to rejoin the group starting from d.
(6) Otherwise, d rejects the join request.

According the first three cases, we can see that the DH node reserves the child location to DA and other DH nodes. From case (4) and (5), we also know that DH and DA nodes have priority over FH nodes when they compete for the same child location.

Procedure JoinDomain(*m*,*n*,*k*)
// Newcomer *n* joins a domain leaded by *m*. *k*=1 or 2.
1: Initialize：*candi_D*←*m*; stack *D* is allocated for storing past *candi_D* nodes.
2: Query *candi_D* to discover all its children. Let Ψ mean the set of *candi_D*, DA and CH children, and FH children if *k*=2. Measure the rtt from each member in Ψ to *n*.
3: Find the nearest member among nodes without *full* label in set Ψ. If all nodes in Ψ are with a *full* label, then pop the top element *p* in *D*, *candi_D*←*p*, go to Line 2.
4: If the nearest member is not the current candidate node, push *candi_D* into the stack *D*, *candi_D*←the nearest member, go to Line 3.
5: Send a JoinRequest (*candi_D*,*k*) request to *candi_D*. If refused, mark *candi _D* with a *full* label and go to Line 3. Otherwise, *candi_D* becomes *n*'s parent node, and mark itself by a DA label if *candi_D*=*m* and *k*=1, or A CH label if *candi_D*≠*m* and *k*=1, or A FH label if *k*=2.

Fig. 4. Join the group to become a domain member or FH

The CM or FH node (denoted by b) accepts join request only if $f'(b) \geq 1$. The DA node (donated by g) responds to a join request sent by t in terms of the following cases:

(1) If $f'(g) \geq 2$, g accepts join request of any type.
(2) If $f'(g)=1$, $\sigma_4(g) \geq 1$ and g receives a JoinRequest(g,1) message, then: g accepts t as its CH child; g randomly selects a FH child of g and tells it to rejoin the group starting from g.
(3) Otherwise, g rejects the join request.

In the application layer multicast, clustering nearby members makes the tree structure congruent to the network topology to some extent, which can effectively improve the performance of the ALM tree. The TCM-based tree building approach tries to divides most of the group members into many domains in terms of the distance, which is helpful for HRALM to cluster the

members. HRALM positions the newcomer by searching the existing tree. The searching procedure stops when it reaches a leaf node, or a node that is closer to the newcomer than the related neighbors. Therefore, we can say that the HRALM tree is topology-aware in some degree.

In HRALM, each host periodically sends the message including current root path to its children, and instantly sends the message when it finds that its root path is changed. Each member *m* also periodically sends the heartbeat messages to keep its neighbor nodes active. When a DH node leaves the group gracefully, it will actively tell the neighbors to cope with its leave. If a member leaves the group without any notification, all the children of the member rejoin the group starting from a closest and active upstream node in its root path. As to the other maintenance procedures, such as structure update, partition recovery, loop detection and resolution, HRALM can use the approaches of some tree-based ALM protocols. We do not care these details in this paper.

5. Loss Recovery of HRALM

In this section, we first introduce the transmission characters of ALM, which is the basis of the design of the loss recovery in HRALM. Then we introduce the related packet and timer types. Finally, we explain the loss recovery solutions for the members in Plane 1 and 2, respectively.

5.1. Transmission Characters of ALM

In IP multicast tree, the loss at a group member (other than the root) has no direct influence on other group members. In contrast, if a node (i.e., group member) in the ALM tree could not receive some correct packet for any reason, all its downstream nodes in the tree would lose the corresponding correct packet because group members take on the forwarding functionality of multicast routers. In ALM, from the root to each member, there is one unique loop-free path along the multicast tree. The member list of this path is called root path [2]. Therefore, there is a high error correlation among the nodes in a root path. Clearly, the group member in multicast session can leave randomly, and has limited capability of forwarding the data because of some reasons (e.g., network congestion and resource exhaust). Therefore the forwarding functionality of the member host is unreliable in ALM.

In a delivery tree (including n+1 nodes) whose maximum node degree is *k*, we can easily conclude that the node at level *i* has at least $n+1-\frac{(k^{i-1}-1)}{k-1}-k^{i-1}$ downstream nodes.

In a given delivery tree, if a node *n* at level *i* loses a correct packet but all its upstream nodes in its root path receive the correct packet, we say that an

interrupt event happens at the node *n,* denoted by $I(n)$. We use $p_j(n)$ to mean the *j*th upstream node in *n*'s root path. For examples, $p_1(n)$ means the parent of node n, and $p_2(n)$ denotes *n*'s grandfather. Assume that the event, that a node cannot correctly receive each packet sent by its parent in the delivery tree, is independent distributed. Then we have:

Lemma 1. The interrupt event happens with higher probability at lower level in the delivery tree. Note that the root is at the lowest level (i.e., level 1).

Proof. Let a node *n* cannot receive the correct packet sent by its parent with probability of α_n, then the interrupt event happens at a node n with probability of $\Pr(I(n))$, $\Pr(I(n)) = \alpha_n \prod_{j=1}^{i-1}(1-\alpha_{p_j(n)})$, where *i* is the level of node *n* in the delivery tree. Therefore we can easily prove the lemma.

According to the above description, we can notice that the interrupt event usually has heavy negative influence on the reliability of the application layer multicast.

5.2. Packet and Timer Types

There are five types of messages (i.e., NACK1, NACK2, NACK1_T, NACK2_A and RECOVERY) and two types of timers (i.e., T_NACK1 and T_NACK2) in the recovery procedure of HRALM, as Table 1 and 2 show.

Table 1. The packet types related to the loss recovery

Packet type	Description
NACK1	Negative acknowledgement sent by a DH
NACK2	Negative acknowledgement sent by a member in Plane 2
NACK1_T	NACK1 acknowledgement packet
NACK2_A	NACK2 acknowledgement packet
RECOVERY	Recovery packet

Table 2. The timer types related to the loss recovery

Timer type	Description
T_NACK1	NACK1 retransmission timer
T_NACK2	NACK2 retransmission timer

The NACK1 and NACK2 messages are the retransmission requests sent by members in Plane 1 and Plane 2, respectively. When a DH node received a NACK1 message from member *m*, it sends the NACK1_T message to tell *m* to restart its T_NACK1 timer if the expected ADU is not in its buffer and it will forward the NACK1 message to other DHs. The NACK1_A message is used

to tell the receiver to contact the next recovery source. HRALM can distinguish the RECOVERY message from the normal data packet by identification in the application data unit. A member forwards the received recovery packet if it has not received the data unit carried by the packet before the arrival of the packet. More detail on the above messages can be seen in the following section.

The time intervals of T_NACK1 and T_NACK2 timers are denoted by t_{NACK1} and t_{NACK2}, respectively. When a member in Plane 1 finds that it has lost a correct ADU, it sends the NACK1 messages to some nodes by the unicast means and starts the T_NACK1 timer. Similarly, a member in Plane 2 sends a NACK2 message to a DH by the unicast means and starts the T_NACK2 timer when the member detects a loss. Let *rtt* (*m*) means the round trip time from the data source to member *m*, then t_{NACK1} and t_{NACK2} each are larger than *rtt*(*m*).

5.3. Loss Recovery for DHs

In this paper, we say that a member is at level *k* if there are (*k*-1) upstream nodes in its root path. In HRALM, each DH *m* at level *k* keeps a recovery list (denoted by $L_{m,k}$).The recovery list saves some member nodes (called recovery neighbors) which are potential loss recovery sources for *m*. Note that the tree root is not included in any recovery list. Let $v(L_{m,k})$ denote the number of recovery neighbors in $L_{m,k}$, then

$$v(L_{m,k})=max\{\text{int}(|\alpha-\log k|), \beta\}\ , \quad (1)$$

where α and β are two configuration parameters. Specifically, α and β denote the upper bound and lower bound of the number of recovery neighbors, respectively. The upper bound is used to confine the maximum number of NACK1 messages in the loss recovery procedure, while the lower bound ensures that the loss recovery for DHs is robust. The default values of α and β are 4 and 2, respectively. According to the rule of hierarchical recovery, Eq. (1) also considers the influence of the level of a member in some degree. In HRALM, the address of a DH is periodically delivered to the group members by the multicast way. Then DH node *m* randomly selects $v(L_{m,k})$ DH nodes as its recovery neighbors in terms of the above address information. Additionally, a DH node periodically sends a heartbeat message to each recovery neighbor to acknowledgement the living of the recovery neighbor. Once finding that some recovery neighbor is not active, the DH node replaces it with an active DH node.

The algorithm shown in Fig. 5 explains the procedure of recovering the loss at a DH node. In this paper, *NACK1*(*n*,*m*) denotes the NACK1 message that is used to tell the receiver of the message to retransmit ADU *n* to member *m*, *NACK1_T*(*n*) means the NACK1_T message that is used to tell the receiver of the message to restart the T_NACK1 timer for ADU *n*, and *RECOVERY*(*n*) represents the RECOVERY message that carries ADU *n*. When member *m* in Plane 1 (i.e., DH *m*) finds that it has lost ADU *n*, it instantly sends a NACK1 message to each recovery neighbor in the recovery list by the unicast means,

for requiring the latter to retransmit ADU *n* to *m*, and starts the T_NACK1 timer.

Procedure RecoverDH(*m*, *n*,*v*($L_{m,k}$))
// Recover the lost ADU *n* of DH *m*.
1: *m* sends *NACK1*(*n*,*m*) to each node in *v*($L_{m,k}$), to request the latter to retransmit ADU *n*; *m* starts the T_NACK1 timer.
2: If *m* receives *NACK1_T*(*n*), it restarts the T_NACK1 timer.
3: If *m* receives *RECOVERY*(*n*) before the T_NACK1 timer expires, the loss at *m* is recovered. Additionally, it duplicates and forwards the message to its children if it has not previously received the ADU.
4: If *m* have not received *RECOVERY*(*n*) before the T_NACK1 timer expires, it sends *NACK1*(*n*,*m*) to the root.

Fig. 5. The recovery algorithm for the DH node

When DH *p* receives *NACK1*(*n*,*m*), it retransmits ADU *n* to *m* if ADU *n* is in its buffer; Otherwise, *p* sends *NACK1*(*n*,*m*) to each recovery neighbor in the recovery list and sends *NACK1_T*(*n*) to *m*. Any member duplicates and forwards received *RECOVERY*(*n*) to all its children if it has not received ADU *n* before the arrival of the recovery packet. In the application layer multicast, the loss at a member results in the same loss at all downstream nodes of the member. Therefore the above solution can actively recover the loss at downstream members. However, the active recovery might produce some repeated ADUs. To address the above problem, the members detect and discard the repeated ADUs.

In the practical application, it is seldom that a DH node sends a NACK1 message but receives no response. For providing complete reliable loss recovery, HRALM also copes with the above situation, i.e., a DH node sends the NACK1 message to the tree root if there is no response to the previous NACK1 message (see Line 4 of the algorithm shown in Fig. 5).

From the algorithm shown in Fig. 5, we can notice that the possible recovery source (i.e., *m*'s recovery neighbors, recovery neighbors of *m*'s recovery neighbors, and so on) exponentially increases in the recovery procedure. Therefore the loss recovery for the DH node is quick and robust.

Fig. 6 illustrates two examples of recovering the loss at a DH node. In Fig. 6a, member *m* sends the NACK message (i.e., *NACK1*(*n*,*m*)) to each node in its recovery list (including n_1, n_2 and n_3). Once receiving *NACK1*(*n*,*m*), node n_3 retransmits ADU *n* to *m* because ADU *n* is in its buffer. In the example, member *m* receives no response from node n_1 and n_2 for some reasons, such as the network congestion and member departure. Fig. 6b depicts another recovery procedure. In this case, n_1 sends *NACK1_T*(*n*) to *m* and sends *NACK1*(*n*,*m*) to each node in its recovery list. Finally, *m* gets *RECOVERY*(*n*) from n_4.

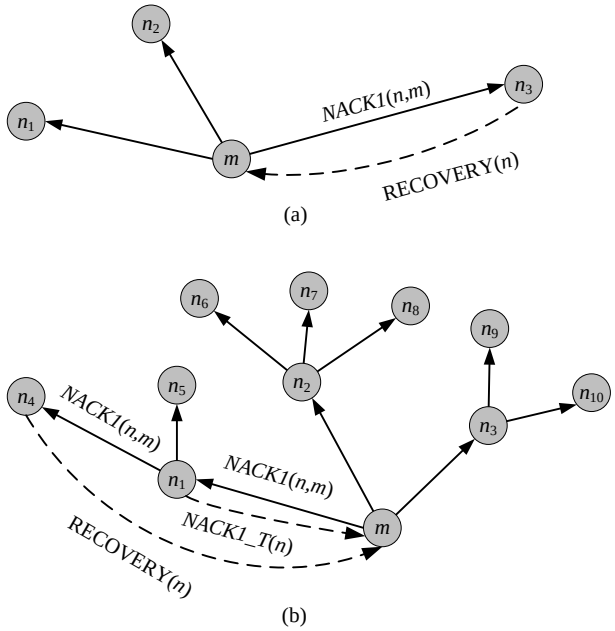


Fig. 6. Examples of recovering the loss at a DH node

5.4. Loss Recovery for Members in Plane 2

As noted previously, only the members in Plane 2 (i.e., DAs, CHs and FHs) can send the NACK2 messages. HRALM employs a multi-round procedure to recover the loss triggered by NACK2. In this situation, a recovery round starts when a member sends a NACK2 message to one of its upstream nodes in the corresponding delivery tree, and ends when the member receives the expected ADU (carried by RECOVERY message) or T_NACK2 timer expires.

Assume that there are $n_r(m)$ nodes between m and the tree root in m's root path, and $n_h(m)$ nodes between m and the first (i.e., closest) DH node in m's root path, then we define a recovery source selection function as

$$U_i(m)=\min\{n_r(m), n_h(m)+i\}\ , \qquad (2)$$

When member m in Plane 1 finds that it has lost a correct ADU (denoted by n), it waits a random time interval between 0 and $rtt(m)$, then sends a NACK2 message (i.e., $NACK2(n,m)$) to the $U_1(m)$th upstream node in m's root path, for requiring the latter to retransmit ADU n, and starts the T_NACK2 timer. When the $U_1(m)$th upstream node receives the NACK2 message, it retransmits ADU n to m if ADU n is in its buffer; Otherwise, it sends a NACK2_A message to m to acknowledge the receipt of the NACK2 message. Once (1) m receives the NACK2_A message before the timer expires or (2) the T_NACK2 timer expires, it sends the NACK2 message to the $U_2(m)$th upstream node in m's root path and restarts the T_NACK2 timer. The above

procedure goes on until *m* receives the expected recovery packet. Fig. 7 gives an example of recovering the loss at the member in Plane 2.

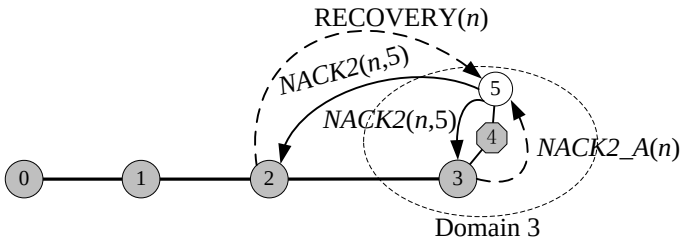


Fig. 7. Example of recovering the loss at the member in Plane 2

According to structure of the HRALM tree, the members in Plane 2 usually are at the middle or bottom of the ALM tree. Therefore the loss recovery operations for members in Plane 2 are effectively reduced by the active recovery mechanism and NACK2 suppression. From Eq. (2), we can see that the recovery source selection is based on a linear function, and each recovery source is a DH node. As noted above, the loss at a DH node can be quickly and robustly recovered. Consequently, the loss at member *m* also can be quickly recovered in most cases.

Since a domain member first sends a NACK2 message to the header of the domain that it belongs to and rechooses the next recovery source by a linear function, the NACK2 explosion problem is effectively alleviated. In addition, the NACK2 suppression further alleviates the NACK2 explosion problem.

6. Simulation Experiments

We used the GT-ITM Generator [30] to generate a 5000-node transit-stub graph as our underlying network topology. Each node represents a router, and the average degree of router nodes was about 3. We also generated 1000 nodes as member hosts, which were connected to stub-domain router nodes randomly. Each stub-domain node connected a host node at most. The fanout of 960 member hosts were assigned by a random value between 2 and 5, and the fanout of other member hosts were assigned by 1. The server was located in a random stub-domain. In the simulations, packets are randomly dropped in each link with probability of an interval μ (called reliability interval). By default, μ=[0.01,0.2], $\lambda=0.25t_{max}$, where t_{max} is the maximum value of rrts between the members and the server. We simulated the related protocols with NS-2 ([31]).

In our experiments, we first used HRALM and LER to build the ALM tree with 1000 receivers, respectively. Table 3 gives the distribution of members of different types in HRALM. From the table, we know that about 10% members are located in Plane 1. Fig. 8 compares the load of the main links, of which

each connects two nodes in different stub-domains, of HRALM and LER trees. Physical link load (stress) means the number of identical copies of a packet that traverse a physical link. As noted previously, LER randomly divides member hosts into several independent planes, which results in that some nearby nodes cannot be well clustered. Therefore LER has higher link load than HRALM (see Fig. 8). In addition, there are more related main links in LER. Note that the related links denote the links that connect the nodes in different stub-domains and transport the data packets of the group application.

Table 3. The distribution of different types of receivers

DHs	DAs	CHs	FHs
109	245	614	32

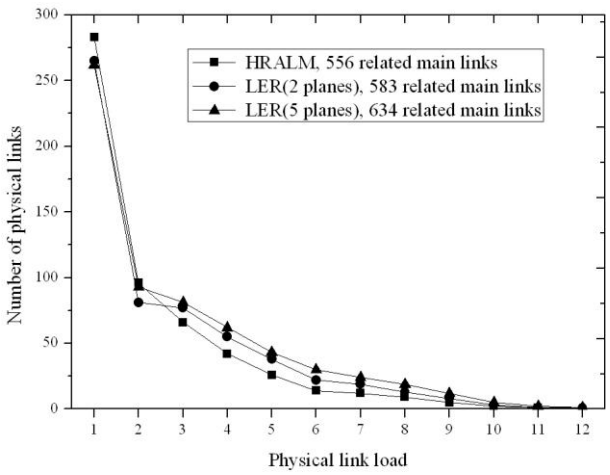


Fig. 8. Physical link load of HRALM and LER

Fig. 9 shows the mean numbers of recovery rounds of HRALM. In each scenario of these experiments, a HRALM tree with 1000 receivers was built, and then 100 ADUs are distributed along the tree with 3 reliability intervals, respectively. In this part, a loss recovery round means a recovery round in the recovery procedure for a loss at the member in Plane 2, or a NACK1 diffusion phase (i.e., the NACK messages are sent by some node for recovering a certain loss) in the recovery procedure for a loss at the member in Plane 1. In particular, the number of loss recovery rounds for a loss is zero if the loss is actively recovered. From Fig. 9, we can notice that the mean number of loss recovery rounds is low in each scenario, which means that HRALM can quickly recover the loss.

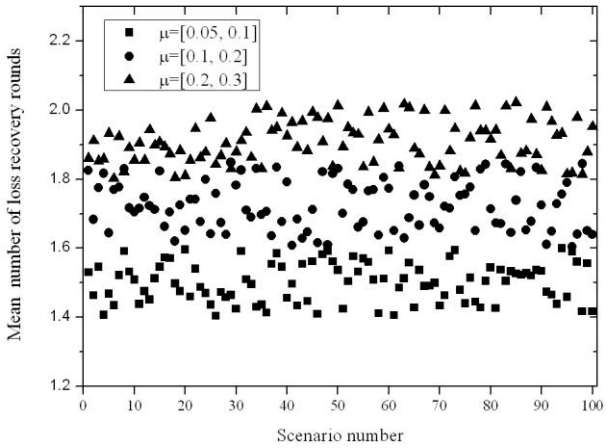


Fig.9. Loss recovery rounds of HRALM

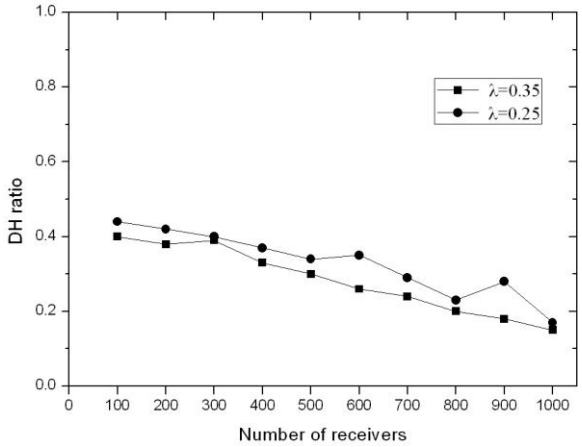


Fig.10. The number of DHs of HRALM

Fig.10 shows the number of DHs in 10 groups with different group sizes. In the figure, DH ratio denotes the ratio of the number of DHs to the number of all members. From the figure, we can see that DH ratio has a dropping trend as the group size grows, which means more and more nodes are clustered with the growth of group size. Since HRALM uses the distance-based domain to contain the domain members, the DH ratio would continue to decrease if more members joined the same group session.

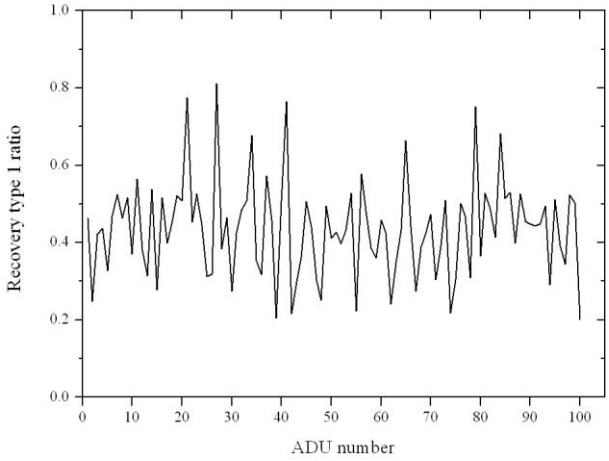


Fig.11. The recovery type 1 ratio of HRALM

In next experiments, we divided the recovery packets into two types, i.e., recovery type 1 and recovery type 2. If a recovery packet is sent to the related members because of the loss recovery launched by a DH node, then the packet is identified by type 1. Otherwise, the recovery packet is identified by type 2. Fig.11 plots the ratio of the number of the recovery packets of type 1 to the number of total recovery packets. From the figure, we can see that the recovery launched by the members in Plane 1 plays an important role through only about 10% group members are DHs.

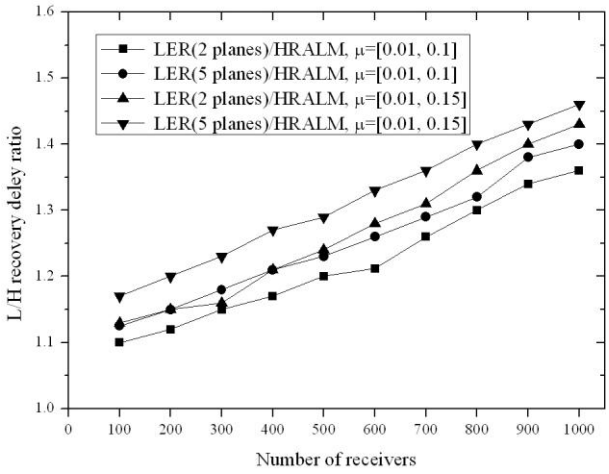


Fig.12. Recovery delay comparison of HRALM and LER

Fig.12 and Fig.13 illustrate the recovery delay and of recovery load HRALM and LER in 10 groups with different group sizes. In these experiments, we used two planes and five planes to build LER multicast tree, respectively. L/H

recovery delay ratio means the ratio of average recovery delay of LER to that of HRALM. Let $l(n)$ mean the number of links that transport the recovery packet n, and $R(A)$ mean the set of recovery packets in solution A. Then L/H recovery load ratio (denoted by *LHLR*) is defined as

$$LHLR = \sum_{n \in R(LER)} l(n) \Big/ \sum_{n \in R(HRALM)} l(n) \quad . \tag{3}$$

According to the above definition, we can notice that *LHLR* can evaluate the load of the loss recovery. From Fig.12, we can see that L/H recovery delay ratio is more than 1 in each group, and that ratio increases as the group size grows. We attribute the desirable performance to quick loss recovery for DHs and the active recovery mechanism. From Fig.13, we notice that L/H recovery load ratio is also more than 1 in each group, and has an increasing trend. Fig.12 and Fig.13 tell us that HRALM can obviously improve the recovery performance.

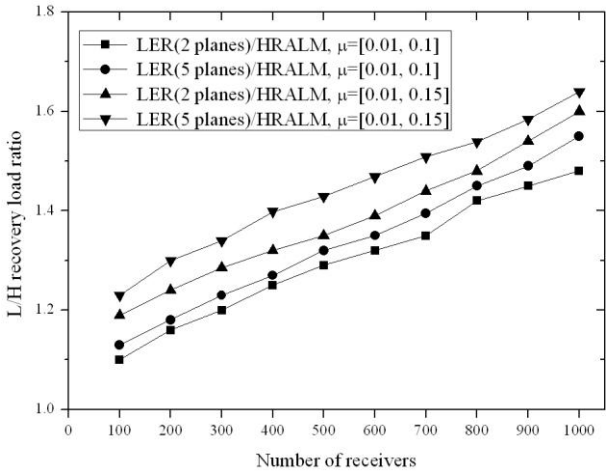


Fig.13. Recovery load comparison of HRALM and LER

7. Conclusion

In this paper, we proposed a two-tiered reliable application layer multicast solution HRALM, which can provide lossless services. In HRALM, most of members belong to the distance-based domains. In the domain, the distance between each member and the domain center is below a given threshold. The distance-based domain can cluster nearby members well. According to the structure of the domain-based tree, HRALM divides the members into four types of members, i.e., domain header (DH), domain agent (DA), common host (CH) and foreign host (FH). The DH node is the center of the corresponding domain, and all the DHs constitute a transport plane, i.e.,

Plane 1. Another plane, named Plane 2, consists of all the DAs, CHs and FHs.

HRALM uses NACK-based mechanism to recover the loss. However, the NACK message is sent to different member (or set of members) in different recovery phase or round because of the unreliability of member hosts. Since the members in Plane 1 are usually in the top of the HRALM tree, HRALM recovers the loss at members in Plane 1 through a quick and robust approach, which ensures that the nodes in Plane 1 are relatively reliable. Based on these relatively reliable nodes, HRALM uses multi-round approach to recover the loss at members in Plane 2. Through the above hierarchical recovery solution, the recovery performance in HRALM is effectively improved.

HRALM also use an active recovery mechanism. In the mechanism, any member duplicates and forwards the received recovery packet to its children if it has not received the data unit carried by the recovery packet before receiving the recovery packet. Since the loss at a member must result in the same loss at downstream node, the above mechanism can effectively improve the recovery performance.

Acknowledgments. This work was supported by the National Natural Science Foundation of China under Grant No. 61070039 and No. 61005029，the National Basic Research Program of China under Grant No. 2009CB320502，the Outstanding Young and Middle-aged Scholars Foundation of Shandong Province of China under Contract No. 2008BS01019.

References

1. Diot, C., Levine, B., Lyles, B., Kassem, H., Balensiefen, D.: Deployment Issues for the IP Multicast Service and Architecture. IEEE Network, 14(1), 78-88. (2000)
2. Zhang, B., Jamin, S., Zhang, L.: Host multicast: A framework for delivering multicast to end users. In Proceedings of IEEE INFORCOM, 1366-1375. (2002)
3. Pendarakis, D., Shi, S., Verma, D., Waldvogel, M.: ALMI: An Application Level Multicast Infrastructure. In Proceedings of the 3rd Usenix Symposium on Internet Technologies & Systems, 49-60. (2001)
4. Wang, F., Xiong, Y. Q., Liu, J. C.: mTreebone: A Collaborative Tree-Mesh Overlay Network for Multicast Video Streaming. IEEE Transactions on Parallel and Distributed Systems, 21(3), 379-392. (2006)
5. Yue, J., Wu, C.: A Trees-Mesh Based Application Layer Multicast Using Collaborative Sub-streams. In Proceedings of the Second International Conference on Future Networks, 29-33. (2010)
6. Wei, X. J., Jia, R. S., Liu, G., Yan, X. H.: Research on P2P-Based Application Layer Multicast Technology for Streaming Media. In Proceedings of the Second International Workshop on Education Technology and Computer Science, 341-345. (2010)
7. Banerjee, S., Kommareddy, C., Kar, K., Bhattacharjee, B., Khuller, S.: Construction of an Efficient Overlay Multicast Infrastructure for Real-Time Applications. In Proceedings of IEEE INFOCOM, (2003)

8. Francis, P.: Yoid: Extending the Multicast Internet Architecture. http://www.aciri.org/yoid
9. Strufe, T., Günter, S., Chang, A.: BCBS: An Efficient Load Balancing Strategy for Cooperative Overlay Live-Streaming. In: IEEE ICC, 1-12. (2006)
10. Li, X. L., Striegel, A. D.: A case for Passive Application Layer Multicast. Computer Networks, 51(11), 3157-3171. (2007)
11. Banerjee, S., Lee, S., Bhattacharjee, B., Srinivasan, A.: Resilient Multicast Using Overlays. IEEE/ACM Transactions on Networking, 14(2), 237-248. (2006)
12. Tian, R. X., Zhang, Q., Xiang, Z., Xiong, Y. Q., Li, X., Zhu, W. W.: Robust and Efficient Path Diversity in Application-Layer Multicast for Video Streaming. IEEE Transactions on Circuits and System for Video Technology, 15(8), 961-972. (2005)
13. Levine, B. N., Paul, S., Garcia-Luna-Aceves, J. J.: Organizing multicast receivers deterministically by parcket-loss correlation. Multimedia Systems, 9(1), 3-14. (2003)
14. Yoon, W., Lee, D., Youn, H. Y.: Comparison of tree-based reliable multicast protocols for many-to-many sessions. IEE Proc., Commun., 15(6): 923-93. (2005)
15. Zhang, X., Li, X., Luo, W., Yan, B.: An Application Layer Multicast Approach Based on Topology-Aware Clustering. Computer Communications, 32(6), 1095-1103. (2009)
16. Simon Wong, K.-F., Gary Chan, S.-H., Wong, W., Zhang, Q., Zhu, W., Zhang, Y.: Lateral Error Recovery for Application-Level Multicast. In: IEEE Infocom, 2708-2718. (2004)
17. Jin, X., Yiu, W. –P. K., Chan, S. –H. G.: Loss Recovery in Application-Layer Multicast. IEEE Multimedia, 15(1), 18-27. (2008)
18. Padmanabhan, V. N., Wang, H. J., Chou, P. A., Sripanidkulchai, K.: Distributing streaming media content using cooperative networking. In: ACM Int. Workshop Network and Operating System Support for Digital Audio and Video (NOSSDAV), 177-186. (2002)
19. Castro, M., Druschel, P., Kermarrec, A.-M., Nandi, A., Rowstron, A., Singh, A.: Splitstream: High-bandwidth content distribution in cooperative environments. In: Int. Workshop Peer-to-Peer Systems (IPTPS), 298-313. (2003)
20. Tian, R., Zhang, Q., Xiang, Z., Xiong, Y., Li, X., Zhu, W.: Robust and efficient path diversity in application-layer multicast for video streaming. IEEE Trans. Circuits Syst. Video Technol., 15(8), 961-972. (2005)
21. Kobayashi, M., Nakayama, H., Ansari, N., Kato, N.: Robust and efficient stream delivery for application layer multicasting in heterogeneous networks. IEEE Trans. Multimedia, 11(1), 166-176 (2009)
22. Alasti, M., Sayrafian-Pour, K., Phremides, A., Farvardin, N.: Multiple description coding in networks with congestion problem. IEEE Trans. Inf. Theory, 47(3), 891-902. (2001)
23. Goyal, V. K.: Multiple description coding: Compression meets the network. IEEE Signal Process. Mag., 18(5), 74-93. (2001)
24. Paul, S., Sabnani, K. K., Lin, J. C., Bhattacharyya, S.: Reliable Multicast Transport Protocol (RMTP). IEEE Journal on Selected Areas in Communications, 15(3), 407-421. (1997)
25. Adamson, B., Bormann, C., Handley, M., Macker, J.: Negative-acknowledgment (NACK)-Oriented Reliable Multicast (NORM) Protocol. IETF RFC 5740. (2009)
26. Wonyong, Y., Dongman, L., hy Youn, Lee, S.: A Combined Group/Tree Approach for Scalable Many-to-many Reliable Multicast. Computer Communications, 29(18), 3863-3876. (2006)
27. Yavatkar, R., Griffioen, J., Sudan, M.: A reliable dissemination protocol for interactive collaborative applications. In: ACM Multimedia, pp.333-344. (1995)

28. Zhang, X., Wang Z., Luo W., Yan Baoping. A Study on Topology-aware Application Layer Multicast Scheme. Journal of Software, 21(8), 2010-2022. (2010) (in Chinese)
29. Handley, M., Floyd, S., Whetten, B., Kermode, R., Vicisano, L., Luby M.:The Reliable Multicast Design Space for Bulk Data Transfer. RFC 2887. (2000)
30. Calvert, K., Zegura, E., Bhattacharjee, S.: How to Model an Internetwork. In: Proceedings of IEEE INFOCOM, 594-602. (1996)
31. The Network Simulator-ns2. Available: http://www.isi.edu/ns-nam/ns.

Xinchang Zhang received the Ph.D. degree from Computer Network Information Center of Chinese Academy of Sciences, Beijing, China, in 2010. Now, he is working at Shandong Computing Center, Shandong Academy of Sciences. His research interests include network protocols and architectures, multicasting, and software testing.

Meihong Yang received the M.S. degrees from the Shandong University. Now, she is the associate technology officer and professor of Shandong Computing Center of Shandong Academy of Sciences. In the recent years, she has received the scientific and technological progress award four times. Her research interests include software engineering and computer network.

Guganggang Geng is a Research Assistant at Computer Network Information Center, Chinese Academy of Sciences. He obtained Ph.D. degree from Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2008. His current research interests include Machine learning, Web Search.

Wanming Luo is an Associate Professor at Computer Network Information Center, Chinese Academy of Sciences. He obtained Ph.D. degree in Computer Science from Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2001. His current research interests include computer networks, performance evaluation, and network security analysis.

Xingfeng Li received the Ph.D. degree from Computer Network Information Center of Chinese Academy of Sciences, Beijing, China, in 2009. Now, he is working at The People's Bank of China. His current research interests include computer networks, system reliability.

Received: December 23, 2010; Accepted: January 20, 2011.

DOI: 10.2298/CSIS100423035J

Multi-Scale Image Semantic Recognition with Hierarchical Visual Vocabulary[1]

Xinghao Jiang[1,2], Tanfeng Sun[1,2], and GuangLei Fu[1]

[1] School of Information Security Engineering
Shanghai Jiao Tong University, Shanghai, 200240, China
{xhjiang, tfsun}@sjtu.edu.cn
[2] Key Lab. of Shanghai Information Security Management and Technology Research,
Shanghai, 200240, China

Abstract. Local features have been proved to be effective in image/video semantic analysis. The BOVW (bag of visual words) scheme can cluster local features to form the visual vocabulary which includes an amount of words, where each word is the center of one clustering feature. The vocabulary is used to recognize the image semantic. In this paper, a new scheme to construct semantic-binding hierarchical visual vocabulary is proposed. Some attributes and relationship of the semantic nodes in the model are discussed. The hierarchical semantic model is used to organize the multi-scale semantic into a level-by-level structure. Experiments are performed based on the LabelMe dataset, the performance of our scheme is evaluated and compared with the traditional BOVW scheme, experimental results demonstrate the efficiency and flexibility of our scheme.

Keywords: local feature, bag of visual words, image semantic analysis, visual vocabulary.

1. Introduction

With the rapid development of Internet and multimedia technology, explosively growing amount of images and videos can be acquired from the web or relevant database. The content-based image/video classification will play more and more important role in the field of images/videos processing. Human can easily figure out different genres of images/videos just by watching them. However, for the computer, it is a quite complicated work to automatically recognize the semantic of a image/video. How to use computer to analyze image semantic has been discussed and researched as a hot topic in this field. The research on image/video semantic analysis is closely

[1] Corresponding author: Dr. Tanfeng Sun.

connected with many applications, such as: content-based image or video retrieval system, the utility in intelligent traffic and safety surveillance, and so on.

A lot of work has been concentrated on some global features extracted from images such as color and texture [1, 2, 3]. An image can be represented by a global feature vector. Then the problem of analyzing image semantic is turned into the problem of supervised classifying. Support Vector Machine (SVM) can be used to judge whether an image belong to one semantic or the other based on amount of training features. Though the use of global feature need only cheap computing cost, its effectiveness is poor and reveals unsatisfied performance.

Local feature has been studied as an improvement on global feature. DoG (Difference-of-Gaussian) [4] is used to detect interest points from image and then SIFT (Scale Invariant Feature Transform) [4] is used to extract a vector of feature from each of those points. Feature is described by the pixel values around the interest point. In this way, an image can be represented as a collection of feature vectors. An easy way to analyze whether image includes some object is to match the feature collection of object image with the feature collection of testing image [5, 6]. Some matching structure can be used in this process to decrease the cost. But it is still not very efficient in recognizing multi-object or complicated semantic.

Recently a new model called BOVW (bag of visual words) which reflected on the BOW (bag of words) model in document retrieval has been discussed widely [7, 8, 9]. BOVW also takes advantage of local feature of image. Like the way BOW works, BOVW can be used to construct a visual vocabulary of an image. The building of visual vocabulary is done by clustering all the feature vectors extracted from the training images. Clustering generated a certain number of cluster centers in feature space. In this way, each cluster center is regarded as a word in visual vocabulary. Each feature vector extracted from image can find its nearest word in vocabulary. Then an image can be represented as a word vector in which each dimension number means that whether the image contains the word. The training image is used to train the SVM for the classifying task. Some details about the BOVW model such as weighting strategy, vocabulary size has been discussed in several papers [10, 11] as well. Though BOVW model has been proved to be more effective on problem of image objects or semantic analysis, it still has at least two drawbacks: 1) the features which are used to construct visual vocabulary have no semantic connection. This leads to the loss of semantic information of visual vocabulary, which the noise feature may have negative influence on the result of image analysis; 2) there is not an efficient structure which can fit large vocabulary. Small semantic analysis may be solved smoothly by the traditional BOVW model, but when there is need for complicated semantic recognition or multi-level semantic analysis, the traditional BOVW model is not enough.

Our new work aims at the drawbacks about the traditional BOVW model mentioned above. The work in [12] proposed the way of semantic-preserving BOVW model. Several codebooks which belong to certain semantic can be

constructed firstly, then image can be analyzed by judging whether any feature extracted from it belonged to any codebook. In our method, a new hierarchical semantic model is proposed, which can be applied in complicated semantic analysis. Based on the hierarchical semantic model, the semantic-binding visual vocabulary tree can be constructed. We define some attributes and relationship of the semantic nodes in the model. The hierarchical semantic model is used to organize the multi-scale semantic into a level-by-level structure. Experiments demonstrate the performance of our scheme is efficiency and flexibility.

The experiments are performed based on the LabelMe image dataset from MIT [13] which contains 11,282 objects from 495 categories. The LabelMe dataset is an online interactive image database, from which users can obtain the annotation of objects in each image. The annotation is very useful for helping us to build the semantic-binding visual vocabulary tree.

The rest of this paper is organized as follows: In Sect. 2, some related work which would be involved in our method is introduced. Sect. 3 presents the definitions of semantic attributes and semantic relationship, and the scheme of building hierarchical semantic model. Sect. 4 discusses the construction of semantic-binding visual vocabulary tree. Sect. 5 gives the method to analyze image semantic using our model and vocabulary tree. Sect. 6 shows the experimental results and analysis. In Sect. 7, the conclusion and some future direction are presented.

2. Related Work

2.1. Sparse Image Interest Point Detecting

Sparse image interest point is compared to the dense image interest point which regards each point of image as the target. Ideal sparse image interest point is scale-invariant, affine-invariant and position-invariant. There are corner-like point detector such as Harris Laplas [14] and blob-like point detector such as LoG (Laplacian of Gaussian). Our work uses the DoG [4] in which the detection process involved not only the image itself but also the neighboring images in the scale space. DoG finds the interest point by determining whether it is a local maximum compared with 26 surrounding points (9 points in pre-scale image, 9 points in post-scale image and 8points in current image) and at the same time the point should be the maximum in it scale curve.

2.2. Local Feature Extraction

Local feature extraction computes on the surrounding pixel values of the interest point and puts out a vector representing local feature. SIFT (scale invariant feature transform) [4] has been regarded as an excellent local feature in image analysis compared to other versions of local feature. In our work, we adopt SIFT to extract local features from images. For each interest point in image, SIFT choose 16 areas around it. The direction of every point in area is calculated out by its surrounding pixels. For all points in each area, all the directions are clustered into 8 bins. In this way, each area has eight numbers meaning the histogram of its direction. For this interest point, the 8-bin histogram of direction from 16 areas forms its final 128-dimensional local feature vector.

2.3. Distance metric learning with contextual constraints

Clustering features in traditional BOVW model use the Euclidean distance to calculate the distance between two feature vectors. Though computing Euclidean distance needs less cost, it lost the contextual information of feature. It is because the Euclidean distance does not take the semantic class of a feature into consideration. Distance metric learning can help to solve this problem. Distance metric learning takes advantage of the contextual constraints of feature. The so-called contextual constraint is the class information of feature [15, 16]. With the contextual constraints, a matrix A can be acquired through training. Then the distance of any two feature vectors can be calculated by the Mahalanobis distance as follows:

$$d_{x.y} = (x-y)^T \times A \times (x-y) \quad (\mathbf{1})$$

where x and y are two feature vectors, A is learnt distance metric.

In our work we used the DCA (Discriminative Component Analysis) distance metric learning [17]. The basic idea of DCA is to learn an optimal data transformation that leads to the optimal distance metric by both maximizing the total variance between the discriminative data class and minimizing the total variance of data instances in the same class. DCA firstly calculates two covariance matrices C_b and C_w which describe the total variance between data of the discriminative class and the total variance of data among the same class respectively. The two matrices are computed as follows:

$$C_b = \frac{1}{n_b} \sum_{j=1}^{n} \sum_{i \in D} (m_j - m_i)(m_j - m_i)^T \quad (\mathbf{2})$$

$$C_w = \frac{1}{n}\sum_{j=1}^{n}\frac{1}{n_j}\sum_{i=1}^{n_j}(x_{ji} - m_j)(x_{ji} - m_j)^T \quad (3)$$

In formulas (2) and (3), $n_b = \sum_{j=1}^{n}|D_j|$, m_j is the mean vector of the *j*-th class, x_{ji} is the *i*-th data instance in *j*-th class, and D_j is the discriminative set in which each element is one of n class that has at least one negative constraint to the *j*-th class. DCA resolves the learning task by the optimization as follows:

$$J(A) = \arg\max_A \frac{|A^T C_b A|}{|A^T C_w A|} \quad (4)$$

In formula (4), A is the optimal transformation matrix to be learned. When A is obtained, the optimal Mahalanobis matrix can be calculated by $M = AA^T$.

3. Hierarchical Semantic Model

A lot of work has been done on understanding image semantic by different kinds of image features, either global or local feature. Less attention is paid to the semantic itself. Our work will give out a new semantic model first which contains some attributes and the relationship between semantics.

3.1. Definitions of Semantic Attributes and Semantic Relationship

We have given some brief introduction about the model in our early work [18]. Hierarchical Semantic Model is used to construct all the image semantics in a semantic space. The constructing process is to place an image semantic into the semantic space and to make correct relationship with other image semantics. When the construction process is done, the semantic model is a multi-layer structure. The upper layer is for bigger image semantic and the lower layer is for smaller image semantic. The ‘big’ and ‘small’ are just comparative terms. Fox example, semantic of ‘wheel’ is a small semantic when it is compared to semantic of ‘car’. But semantic ‘car’ is not big enough if you take semantic of ‘street’ into consideration. Actually what is more important is not whether a semantic is big or small, but is the relationship between different semantics. Just like the example mentioned above, ‘car’ should have ‘wheels’, and probably there are many cars on the ‘street’.

First we give out some definition of semantic attributes here. We classify all the semantic into two classes. One class is called the 'combination semantic' and the other is called 'singleton semantic'. Some notations are be used here: 1) σ stands for the scale or granularity of the semantic; 2) $\sum$ stands for the combination of several semantic; 3) $\bigcup$ stands for the union of several semantic or semantic set.

Definition 1. Singleton semantic: singleton semantic describes some simple semantic which has no necessity to be destructed again. An example of singleton semantic is semantic of 'car'. You still can destruct the semantic of 'car' into semantic of 'wheel' or semantic of 'windscreen'. But 'wheel' and 'windscreen' is too simple to form a visual vocabulary individually. This is also what 'no necessity' stands for.

Definition 2. Combination semantic: combination semantic describes some comparatively complicated semantic which are formed by the combination or union of several smaller semantics. An example of combination semantic is semantic of 'street'. Semantic of 'street' can be composed of the semantic of 'road' and semantic of 'car' or semantic of 'house' and so on. If S stands for combination semantic, s stands for the semantic which $\sigma(s) < \sigma(S)$, then $S = \bigcup_{i=1}^{n} ss_i, ss_i = \sum_{k=1}^{m} s_k$.

Then we will discuss the definition of relationship between semantics. There are mainly two kinds of relationship between image semantics: relationship of combination and relation of belonging-to.

Definition 3. Relationship of combination: relationship of combination describes relationship between some smaller semantic and some bigger semantic. All smaller semantic make up the bigger semantic. An example of this kind of relationship is semantic of 'street' (bigger semantic) and semantic of 'car', semantic of 'road', semantic of 'house' (three smaller semantics). Those three smaller semantic form the bigger semantic of 'street'. If S stands for up-level semantic and s stands for the down-level semantic, then the relation ship of combination can be described by $S = \sum_{i=1}^{n} s_i$.

Definition 4. Relationship of belonging to: relationship of belonging to also describes relationship between some smaller semantic and some bigger semantic. The difference from the relationship of combination is that bigger semantic does not need all the smaller semantic. An example is semantic of 'vehicle' (bigger semantic) and smaller semantic of 'car' and smaller semantic of 'truck'. Semantic of 'car' belongs to semantic of 'vehicle' no matter whether there exists semantic of 'truck'. If S stands for up-level semantic and s stands for the down-level semantic, then the relation ship of combination can be described by $S = \bigcup_{i=1}^{n} s_i$.

The following three more definitions are for the relational attributes of the semantic.

Definition 5. Relationship of mutual exclusion: relationship of mutual exclusion describes the relation of two semantics which can't be co-existed. An example is semantic of 'street' and semantic of 'classroom'.

Definition 6. Required semantic: When several small semantics combine into a bigger semantic, some small semantic must be in this combination and such kind of semantic is called 'required'. For example, semantic of 'street' can be combined by semantics of 'road', 'car', 'house', 'walking person' and so on. 'road', 'house' should be two required semantic. Actually whether a semantic is required is closely connected to the application demand. Details will be explained latter.

Definition 7. Optional semantic: this is compared to the definition of required semantic. That is the semantic which may or may not exist in the combination of a bigger semantic.

Those definitions of semantic attributes and semantic relationship are used in the construction of hierarchical semantic model which would be discussed in the following part. Semantic has its own structure and order and our work does take advantage of such kind of order and structure to recognize image semantic.

3.2. Construction of Hierarchical Semantic Model

One different point from the traditional BOVW is that our visual vocabulary tree is bound to certain semantics. In other words, the vocabulary tree must be constructed to match with a semantic model. This also means that before construction of a useful semantic-binding visual vocabulary tree, a hierarchical semantic model should be constructed first. The model we discussed above is an abstract model. If we want to apply this model into practice, we should connect it with some concrete semantics.

For construction of our model, the main task is to decompose a bigger semantic into several smaller semantic iteratively until it comes to some level of singleton semantic. As discussed before, it is not necessary to decompose singleton semantic any more. When all the decomposition has been finished, we shall define the attributes of every semantic node in the hierarchical model and the relationship between any two semantic which are connected with each other. The definitions of attributes and relationship have been introduced in Sect. 3.1. When all the work has been done, a hierarchical semantic model has been successfully constructed. Fig. 1 describes this model.

Fig.1 shows an abstract model of hierarchical semantic. In Fig. 1, top level semantic is the biggest semantic in this model. And there are three combination semantics which maintain the 'relationship of belonging to' with its upper level semantic. For each combination semantic, there are two singleton semantics below them, and the singleton semantics combine into the upper level semantic with the relationship of 'combination'. The number

list like 1, 1 or 1,1,1 in figure just labels the position of relevant semantic node.

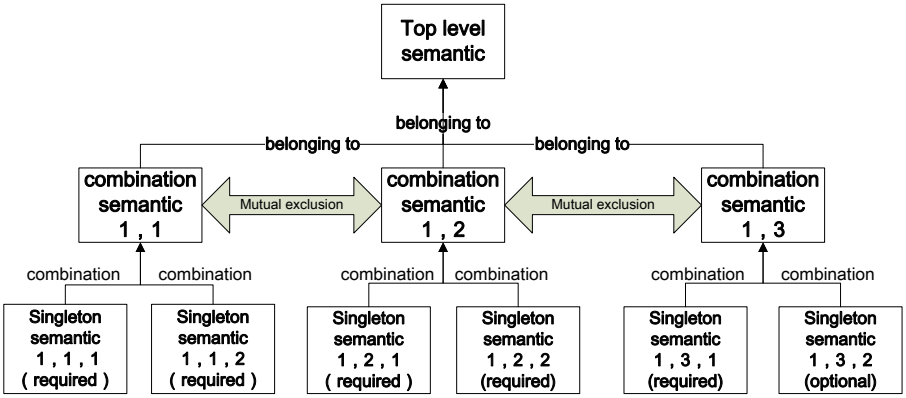


Fig.1. The hierarchical semantic model

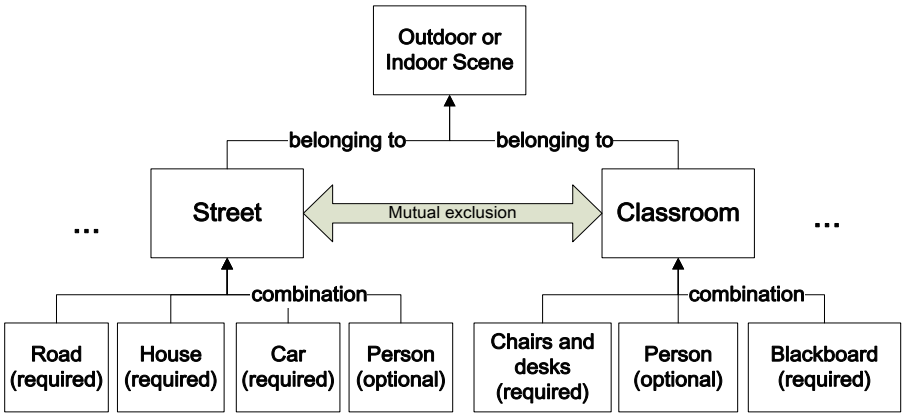


Fig.2. An example of hierarchical semantic model with concrete classification demand

When we put the hierarchical semantic model into practice, concrete classification demand should be taken into account. Fig. 2 shows an example of hierarchical semantic model with application demand, we should think about what semantics need to be recognized and analyzed, what singleton semantics are required and what else are optional. In Fig.2, each semantic node is connected to a concrete semantic. The top semantic is ‘outdoor’ or ‘indoor’ scene. And there are two combination semantics which are ‘street’ and ‘classroom’ belonging to this top semantic. Each of the combination semantic is made up of several singleton semantics. Semantic of ‘street’ combined by singleton semantic of ‘road’, ‘house’, ‘car’ and ‘walking person’ and semantic of ‘classroom combined by singleton semantic of ‘chairs and desks’, ’students’ and ‘blackboard’.

We can see that the singleton semantic of ‘person’ is included both in semantic of ‘street’ and of ‘classroom’ in Fig.2. This sometimes happens especially in some large semantic space situation. Singleton semantic is just like part which always used to make up the large up-level semantic. So the same singleton semantic is very likely to be used in several different large semantic.

For the situation of large semantic space, single hierarchical semantic model maybe is not enough to cover the whole semantic space. We can make extension for the model proposed above. Several models can be built with the certain semantic spaces, so a ‘forest’ can be formed. Each ‘tree’ of this ‘forest’ stands for a united sub semantic space and all ‘trees’ stand for the whole semantic space. One virtual root node can be made to take the charge of all the ‘trees’.

4. Construction of Semantic-binding Visual Vocabulary

The objective for building such a hierarchical semantic model is to make a template on which a visual vocabulary tree can be constructed. As we discussed in Sect. 3, the decomposition of semantic is a top-to-bottom process. On the contrary, the process of constructing a visual vocabulary tree is from bottom to top.

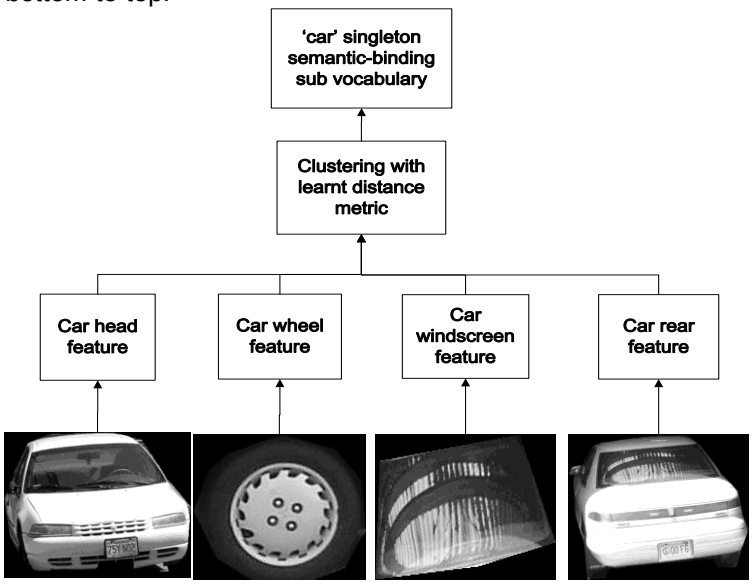


Fig.3. Construction singleton semantic-binding sub vocabulary of ‘car’

Fig.1 and Fig.2 elaborate what the hierarchical semantic model looks like. Actually when we construct a vocabulary tree, each node in the tree is a sub

vocabulary of which semantic is responding to that in the model. Followings are the main steps for constructing a semantic-binding visual vocabulary tree.

Step 1. First, we build those singleton semantic-binding sub vocabularies which are located in the bottom level in hierarchical semantic model. For each singleton semantic node, we collect the images that represent this semantic. SIFT feature is extracted from those images and distance metric is learnt by the contextual information of feature. The contextual information here means different feature comes from different class into one semantic. Take semantic of 'car' as an example. A 'car' can product features from its 'wheel' or its 'windscreen' or its 'body'. When we obtained the distance metric, we cluster features into vocabulary using the learnt metric to compute feature distance. Here k-means algorithm is adopted for clustering. Fig. 3 describes this process briefly, i.e. the features to train the vocabulary bound with semantic of 'car' are taken from different parts of the car, such as 'wheel', 'windscreen', 'car head' and 'car rear'. In this way, a singleton semantic binding vocabulary is successfully built.

Now we have a sub vocabulary of K words (if we set K as the number of clusters in clustering step), and we also should calculate out radius of each word as following formula:

$$r_i = \frac{\sum_{j=1}^{n_i} (\left| x_{ij} - c_i \right|_A)}{n_i} \tag{5}$$

In formula (5), r_i is the word radius for the *i*-th word in a certain semantic binding vocabulary. n_i is the number of the feature vectors belonging to this word. c_i is the clustering center point vector and x_{ij} is each feature vector. A is the learnt distance matrix. We calculate the word radius by means of the sum of the distance between feature and center point. Since not all the features we take to train the semantic binding vocabulary is totally correct, there still may be some noise features, so averaging can decrease the negative influence from those noise features.

The radius of the whole vocabulary is calculated out as follows:

$$R = \max(\left| x_i - vc \right|_A) \tag{6}$$

In formula (6), *R* is the radius of whole vocabulary, x_i is each feature in the vocabulary, VC is the vocabulary center point which equals to the mean of all the word center points. And the longest distance between the features and vocabulary center points is defined as vocabulary radius.

Step 2. After all the singleton semantic vocabulary has been built, we can build the sub vocabulary responding to the up-level semantic in the model.

In this step, we do not need to extract new feature. The features used to build up-level semantic-binding sub vocabulary are formed by features of the semantics combined into it or the semantics belonging to it. And the contextual information here is down-level semantic which the feature comes from. With the feature and contextual information, we can acquire the learnt distance metric. And in the same way, up-level vocabulary can be obtained by clustering the features with the learnt distance metric. Fig. 4 shows us the process to construct the up-level semantic-binding vocabulary from down-level semantic binding vocabularies.

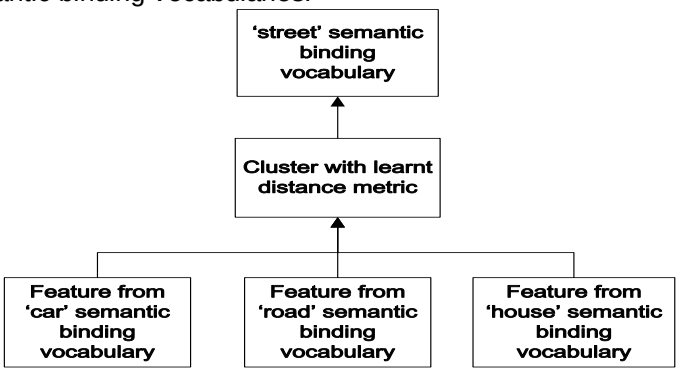


Fig.4. Construct the up-level vocabulary from down-level vocabularies

Step 3. We can build the other sub vocabulary iteratively from bottom semantic to top semantic just in the way Step 2 shows us.

The above steps show how to build a semantic binding visual vocabulary tree. The process makes sure that each sub vocabulary in the tree is bound to a certain semantic. It can be easily proved because in the hierarchical model, the up-level semantic is formed by its down-level semantics. And in the process of building the vocabulary tree, all the features in the down-level semantics run into the up-level semantic eventually. According to the extent of semantic, those features belong to up-level semantic naturally.

Just like the traditional BOVW model, the vocabulary size is an important factor which influents performance greatly. We give out a solution to decide the size of each semantic binding sub vocabulary. The size of a vocabulary is proportional to its complexity, i.e. the more complex it is, the bigger size it has. We use a randomization method to know the complexity and to decide the size.

The features of each semantic binding vocabulary are taken from some different class. As we known, the feature of singleton semantic vocabulary is taken from different parts of the singleton semantic object, and the feature of combination semantic vocabulary is taken from different down-level semantic binding vocabulary. We define ∂ here, and $0<\partial<1$. For a vocabulary which we want to decide its size, we take ∂N_i feature vectors from each of its down-level vocabulary randomly. Where r is the number of down-level

vocabularies, N_i means the number of feature vectors of the *i*-th down-level vocabulary. So we can get $N = \sum_{i=1}^{r} \partial N$ feature vectors and we compute its complexity as follows:

$$D = \frac{\sum_{i=1}^{N} \sum_{j>i}^{j \le N} \left| x_i - x_j \right|_A}{N(N-1)} \quad (7)$$

$$C = \frac{D}{2R} \quad (8)$$

The formula (8) shows that we compute the complexity of the vocabulary by those randomly picked feature. A vocabulary is more complex when its average distance between feature vectors is bigger. Where R is the radius of vocabulary, D is the mean distance among those picked feature and C is the complexity for the vocabulary (0< C <1). We can use C to decide the size of vocabulary as follows:

$$size = C \times SMAX \quad (9)$$

In formula (9), $SMAX$ stands for the largest size of all the vocabularies.

5. Analyzing Image Semantic with Semantic Binding Visual Vocabulary

Just like the traditional BOVW model, a built vocabulary is used to analyze the image features and to perform the final words histogram. In our method, we take the similar way to analyze image semantic. The big difference in our method is that we do not use SVM to classify which semantic image should belong to. Instead we analyze the semantic of image by what sub vocabulary or its semantic the image possessed. The followings are the steps to analyze image semantic based on the semantic binding visual vocabulary:

Step 1. Detect the interest points in test image by DoG method, and the SIFT feature vector is extracted for each interest point.

Step 2. For each SIFT feature vector extracted from test image, we match it with sub vocabularies in visual vocabulary tree from top to bottom. For a certain sub vocabulary, the method to judge whether a feature belong to this vocabulary is described as follows:

$$\bigcup_{\substack{c_i \in vocabulary \\ f_j \in vocabulary}} \delta(\left| f_j - c_i \right|_A < r_i) = 1 \quad (10)$$

In formula (10), δ is the function: when its input argument is true, then its output is 1; when argument is false, then its output is 0. f_j is SIFT feature extracted from image, c_i is *i*-th word centers of the vocabulary, r_i is the word radius for the *i*-th word, and A is the learnt distance matrix for the vocabulary. If the result of formula (10) is 1, f_j drops into the vocabulary.

Actually we can adopt two different strategies. Strategy 0: a feature can drop down into any number of vocabularies on each level. Under such condition, the above formula is used to judge which vocabularies the feature drops into; Strategy 1: a feature can only drop down into one vocabulary on each level. Under such condition, if the above formula shows that a feature may be in several vocabularies on each level, then the feature is discarded as an unstable feature. The comparison will be revealed in experiments (Sect. 6.3) on these two strategies.

Step 3. Match each SIFT feature with top semantic binding sub vocabulary in the way as Step 2. For the feature belonging to this sub vocabulary, we match it to each of the down-level sub vocabulary. This process works iteratively until the feature reaches the singleton semantic binding sub vocabulary or until the feature is discarded for it belongs to no sub vocabulary.

Step 4. Do step 3 on all the SIFT feature vectors extracted from test image. For each bottom-level singleton semantic binding sub vocabulary, we know if it contains any feature. If one singleton semantic contains feature extracted from test image, we say the image possesses this singleton semantic.

Step 5. Now we know what singleton semantic the test image possesses. Based on the hierarchical semantic model we have built, when we know what the down-level semantic the test image possesses, we can know what up-level semantic the test image possesses. We do this semantic aggregation work from bottom (singleton semantic) to top (the biggest semantic). After this process is finished, we can know what semantic the test image has in each level of the hierarchical semantic model. For the example in Fig. 2, If a test image has singleton semantic of ‘car’, ‘house’ and ‘road’, then we can say the image also has semantic of ‘street’. And further we can say the image has a scene semantic.

Step 6. When Step 5 is done, we can have the knowledge of the semantic of image from big to small scale. Here big scale means what combination semantic the image has, and the small scale means what the singleton semantic the image has. One point should be paid attention to here is that if a image gets the two semantics which maintain the relationship of mutual exclusion, then this image should not have either of the both semantics.

6. Experimental Results

We evaluate our model and algorithm from different scales of semantic on the test images. For one test image, we can also evaluate the performance of our work by the accuracy of analysis results on each level semantic. And then the performance of our scheme is evaluated and compared with the traditional BOVW scheme.

6.1. Dataset for Experiments

The experiments are carried out on the dataset provided by LabelMe project from MIT [13]. LabelMe is an image dataset in which each image has a responding annotation file. The annotation file annotates objects of different semantic in the image. So we can collect large number of training material of certain semantic from LabelMe dataset. LabelMe dataset also includes a wide range of image categories which totally covers 11,281 objects from 495 categories.

6.2. Experiment Setting

Two combination semantics are chosen for the experiments. One is outdoor semantic: the semantic of ‘street’. The other is indoor semantic: the semantic of ‘office’. Actually our model can be applied to any combination semantic as long as the semantic can be decomposed in the way we introduced in Sect. 3. The hierarchical semantic model for the experiments should be constructed first, as shown in Fig. 5.

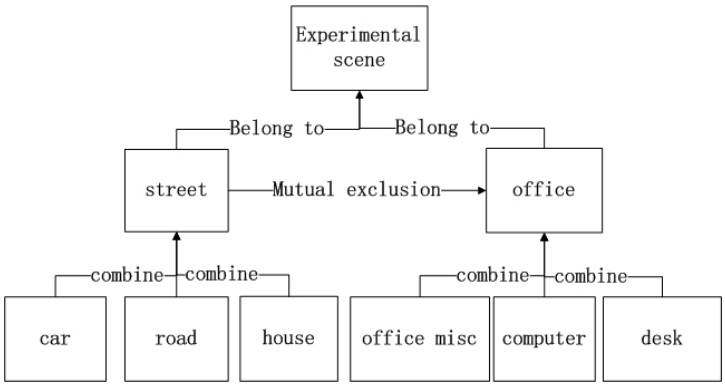


Fig.5. The hierarchical semantic model for our experiments

Fig. 5 shows the semantic model for our experiments. Our experiments aim at two combination semantic ‘street’ and ‘office’, both of them belong to

the top-level semantic. The ‘street’ semantic and ‘office’ semantic have relationship of mutual exclusion between them. For ‘street’ semantic, it is formed by ‘car’, ‘street’ and ‘house’ semantic. All of the three singleton semantics are ‘required’ to their up-level semantic. For ‘office’ semantic, it is formed by ‘office miscellaneous’, ‘computer’ and ‘desk’ semantic. And all of those three semantics are also ‘required’ to their up-level semantic.

The semantic binding visual vocabulary tree is constructed based on the hierarchical semantic model in Fig. 5. Some training images of certain singleton semantic are collected from labelMe dataset first. Fig. 6 describes this process.

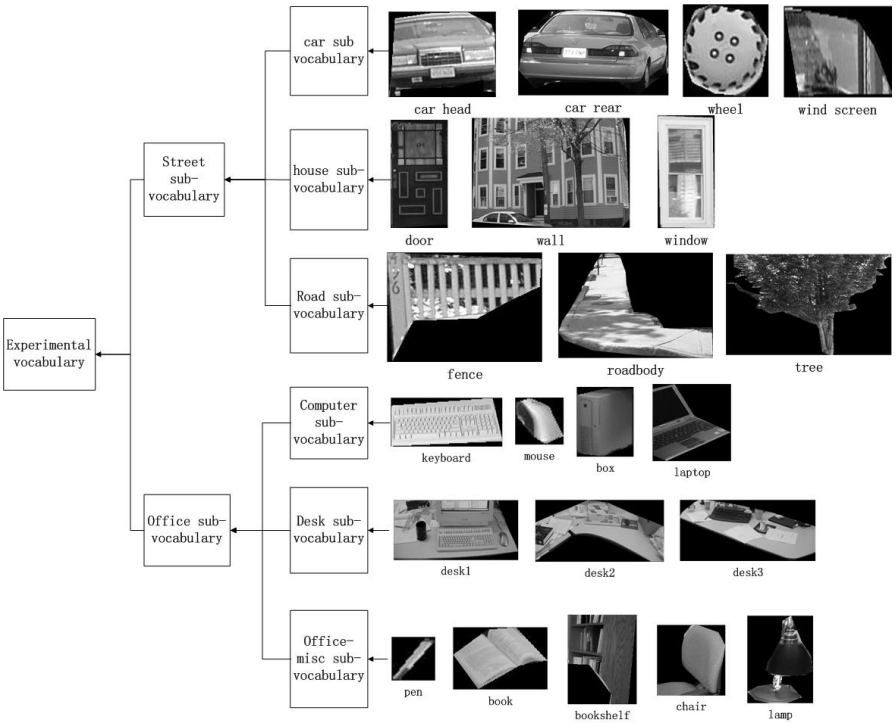


Fig.6. Construction for semantic binding vocabulary tree for experiments

Fig. 6 shows that each bottom-level semantic-binding sub vocabulary is trained by the image of responding semantics. Actually the image shown in Fig. 6 for each singleton semantic vocabulary is from different parts of the object. Just as discussed in Sect. 4, different parts can provide us with the contextual information which is useful in distance metric learning. In the experiment, we learn the distance matrix for each of the six singleton semantic binding vocabulary with the contextual information. Three of the singleton semantic binding vocabularies (‘car’, ‘road’, ‘house’) aggregate to their up-level vocabulary (‘street’), and the other three (‘office misc’, ‘desk’,

'computer') aggregate to the other up-level vocabulary ('office'). The aggregation process has been discussed in details in Sect. 4.

In our experiment, the size of each sub vocabulary is determined in the vocabulary tree according to the complexity of the vocabulary itself. The method has been discussed in Sect. 4, and Table 1 describes the details of the size of each sub vocabulary.

Table 1. The size of each sub vocabulary in the vocabulary tree of our experiment

Top semantic vocabulary (size): 500					
'street' vocabulary (size): 400			'office' vocabulary (size): 300		
'car': 200	'house' : 300	'road' : 300	'office misc' : 300	'computer' : 200	'desk' : 200

For each test image, we extract the SIFT features for each interest point which is detected by DoG method from the image. For each SIFT feature, we analyze it with the vocabulary tree from top to bottom. In Sect. 5, we proposed two strategies in judging whether a feature drops in a vocabulary. One is that in each level a feature can drop in several sub vocabulary, the other is that in each level a feature can only drop in one sub vocabulary and otherwise the feature is discard. We will give a performance comparison in the experiment results.

When all the features from test image are analyzed by the whole vocabulary tree, we recognize the image semantic from the bottom singleton semantic, and gradually to up-level combination semantic. To evaluate our model and method in details, we can give the performance for each level semantic in our experimental model. For each semantic node in the hierarchical model, we evaluate the accuracy of our method as follows:

$$precision_i = \frac{N_{correct}}{N_{test}} \quad (\mathbf{11})$$

$$recall_i = \frac{N_{correct}}{N_{truth}} \quad (\mathbf{12})$$

In formula (11) and (12), i means the *i*-th semantic node in the experimental model, $N_{correct}$ stands for the number of test images which are recognized correctly, N_{test} is the number of total test images, N_{truth} is the number of test images which really have the certain semantic. We give the evaluation for accuracy of each semantic node by two strategies in experiment results.

6.3. Experiment results

In the experiments, totally 1000 test images are chosen from LableMe dataset, in which 500 images of them match the semantics in the experimental model and the other 500 do not match. 'Match' here means the image can match any node semantic in the experimental model. We hope such kind of composition can make the testing more standard and convincing.

We evaluated the performance of our method on every semantic node in experiment hierarchical semantic model with two feature dropping strategies (discussed in Sect. 5). Strategy 0 means that a feature vector can drop into any sub vocabulary in one level, Strategy 1 means that a feature vector can drop into only one sub vocabulary in the level (the multi-dropping feature be discard as unstable feature). Table 2 and Table 3 show the accuracy of precision and accuracy of recall about each semantic. There are 6 singleton semantics ('car', 'house', 'road', 'office misc', 'computer', 'desk') and 2 combination semantics ('street', 'office').

Table 2. Accuracy of each semantic node in experiment model with strategy 0

Feature dropping strategy 0				
semantic	'car'	'house'	'road'	'**street**'
precision	0.74	0.82	0.73	0.70
recall	0.79	0.85	0.75	0.72
semantic	'misc'	'computer'	'desk'	'**office**'
precision	0.79	0.78	0.82	0.78
recall	0.82	0.83	0.73	0.69

Table 3. Accuracy of each semantic node in experiment model with strategy 1

Feature dropping strategy 1				
semantic	'car'	'house'	'road'	**'street'**
precision	0.72	0.72	0.79	0.66
recall	0.77	0.76	0.75	0.67
semantic	'misc'	'computer'	'desk'	'**office**'
precision	0.64	0.66	0.68	0.64
recall	0.68	0.71	0.69	0.61

From Table 2 and Table 3, we can observe that the accuracy (precision or recall) of the singleton semantic is higher than that of the combination semantic. In Table 2 , the precisions of 'car', 'house', 'road' are 0.74,0.82 and 0.73, and all of them are higher than the precision of 'street' (0.70). The precisions of 'office misc', 'computer', 'desk' are 0.79, 0.78 and 0.82, and all of them are also higher than that of 'office' (0.78). So is the recall. The reason is the confirmation of a combination semantic needs all the confirmation of its down-level required semantic. In our experiment model, if an image

possesses semantic ‘street’, it must possess semantic ‘car’, ‘house’ and ‘road’. This leads to the combination semantic (‘street’ or ‘office’) with lower accuracy than the singleton semantic. Compare the results of Table 2 with Table 3, the accuracy of semantic node in Table 3 (with Strategy 1) is lower than that of the responding semantic node in Table 2 (with Strategy 0). The precision of semantic ‘car’ in Table 2 is 0.74 and that in Table 3 decreased to 0.72. The precision of semantic ‘office’ in Table 2 is 0.78 and that in Table 3 decreased to 0.64. The reason is that Strategy 1 has more limits than Strategy 0, so the unstable feature should be discarded according to Strategy 1 (discussed in Sect. 5), thus the number of features dropping into bottom-level sub vocabulary would decrease. Fig. 7 shows the accuracy results with comparison in chart.

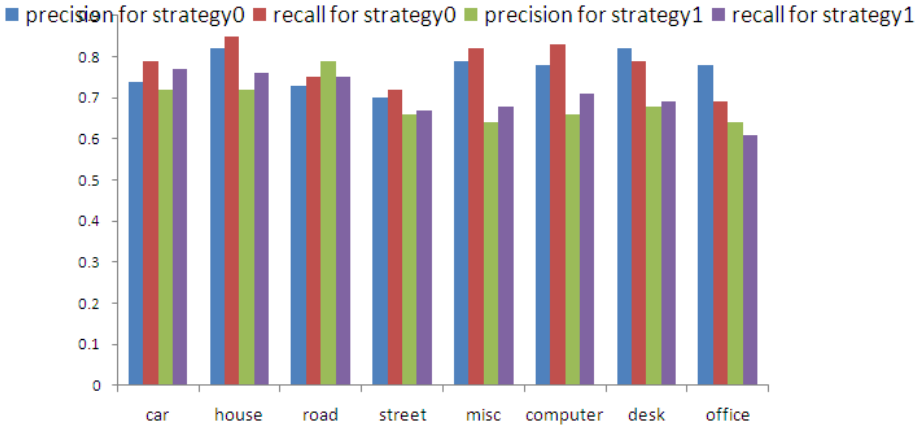


Fig. 7. Accuracy on each semantic node with two strategies

In Fig. 7, X-axis is semantic and Y-axis is accuracy. We can clearly see that the accuracy on singleton semantic is to some extent satisfactory. But the accuracy on combination semantic, especially with strategy 1, has still space to make improvement. Actually the dataset for training and distance metric learning used in vocabulary construction are two important factors in the running of the whole model. In our experimental results, semantic ‘house’ and ‘desk’ get higher accuracy (0.82 and 0.82 individually) since the training dataset for those two sub vocabulary has larger complexity than others. This may inspire us that the complexity and discriminative of the training data can impact the effect of the vocabulary. And the DCA (Discriminative Component Analysis) we used in our model can be replaced by a better distance metric learning algorithm which adapts to our method. This will be studied in our future work.

In order to compare the performance of our model with that of traditional BOVW model, we perform the comparison experiments with BOVW model on the same dataset as well. We constructed the codebooks of BOVW model on the training data from images of semantic ‘street’ or semantic ‘office’.

SIFT features are extracted from all of those training images. And all the features are clustered into 500 clusters. Then a codebook of 500-word size is generated. KNN (K-Nearest Neighbor) algorithm is used to find the nearest word in the codebook. By using the trained SVM, the test images between the semantic 'street' and 'office' can be classified. Table 4 shows the performance comparison between our model and BOVW model.

Table 4. Comparison between our model and BOVW model

Method	Semantic 'street'	Semantic 'office'
BOVW Model (precision)	0.62	0.55
Our method with Strategy 0 (precision)	0.70	0.78
Our method with Strategy 1 (precision)	0.66	0.64

Table 4 shows that even in Strategy 1, the accuracy of our method is still higher than that of traditional BOVW model (semantic 'street' is 4 percentage points higher and semantic 'office' is 9 percentage points higher), which reveal that our model can work effectively in image semantic recognition.

Besides the higher accuracy, our scheme can understand and analyze image semantic in a much more flexible way. Our scheme can analyze images and get the semantic recognition on each semantic node. But for BOVW it is necessary to do the classification for each semantic.

7. Conclusion

In this paper, a hierarchical semantic model is proposed. The hierarchical semantic model is used to organize the multi-scale semantic into a level-by-level structure. The attributes and relationship of the semantic node in the model are defined first. Those definitions are very useful in constructing the hierarchical image semantic model. We also discuss how to construct a semantic-binding hierarchical visual vocabulary tree based on the built hierarchical semantic model. Each sub vocabulary node in the tree is bound to a certain semantic. The semantic binding vocabulary helps to filter out the noise feature and refine the performance. Then the procedure of analyzing image semantic with semantic binding visual vocabulary is described in detail. And two feature dropping strategies are discussed. Experiments are performed based on the LabelMe dataset, the performance of our scheme is evaluated and compared with the traditional BOVW scheme. The experimental results demonstrate the efficiency and flexibility of our scheme. Our model can help to understand and analyze image semantic in a flexible multi-resolution way, and get the semantic recognition on each semantic node. But for traditional BOVW model, it is necessary to do the classification

for each semantic. Our future work will focus on improving the performance of our method, choosing the proper distance learning metric algorithm and the applying in image retrieval system.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (No. 60802057, 61071153), Sponsored by Shanghai Rising-Star Program (10QA1403700), Program for New Century Excellent Talents in University (NCET-10-0569).

Reference

1. Niblack, W., Barber, R., Equitz, W., (ed.): The QBIC project: Querying images by content using color, texture, and shape. In Proceedings of SPIE on Storage and Retrieval for Image and Video Databases, 173-187. (1993)
2. Carneiro, G., and Vasconcelos, N.: Formulating semantic image annotation as a supervised learning problem. In Proceedings of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 163-168. (2005)
3. Fan, J., Gao, Y., Luo, H.: Multi-level annotation of natural scenes using dominant image components and semantic concepts. In Proceedings of the 12th annual ACM international conference on Multimedia, 540-547. (2004)
4. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, Vol. 60, 91-110. (2004)
5. Barnard, K., Duygulu, P., Forsyth, D., Freitas, N.D., (ed.): Matching words and pictures. Journal of Machine learning Research, 1107-1135. (2003)
6. Jin R., Chai, J.Y., Si, L., Effective automatic image annotation via a coherent language model and active learning. In Proceedings of the 12th annual ACM international conference on Multimedia, 892-899. (2004)
7. Wu, L., Hu, Y., Li, M., Yu, N., Hua, X.S.: Scale-Invariant visual language modeling for object categorization. IEEE Transactions on Multimedia, Vol.11, No. 2, 286-294. (2009)
8. Tirilly, P., Claveau, V., Gros, P.: Language modeling for bag-of-visual words image categorization. In Proceedings of 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 249-258. (2008)
9. Yang, L., Jin, R., Sukthankar, R., Jurie, F.: Unifying discriminative visual codebook generation with classifier training for object category recognition. In Proceedings of 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1-8. (2008)
10. Jiang, Y.G., Ngo, C.W., Yang, J.: Toward optimal bag-of-features for object categorization and semantic video retrieval. In Proceedings of the 6th ACM international conference on Image and video retrieval, 494-501. (2007)
11. Yang, J., Jiang, Y.G., Hauptmann, A.G., Ngo, C.W.: Evaluating bag-of-visual-words representations in scene classification. In Proceedings of the international workshop on Workshop on multimedia information retrieval, 197-206. (2007)
12. Wu, L., Hoi, S.C.H., Yu, N.H.: Semantics-preserving bag-of-words models for efficient image annotation. In Proceedings of the First ACM workshop on Large-scale multimedia retrieval and mining. 19-26. (2009)
13. Russel, B.C, Torralba, A., Murphy, K.P., Freeman, W.T.: Labelme: A database and web-based tool for image annotation. International Journal of Computer Vision, Vol. 77(1-3), 157-173. (2008)

14. Leibe, B., Schiele, B.: Scale-Invariant Object Categorization Using a Scale-Adaptive Mean-Shift Search. Pattern Recognition. Lecture Notes in Computer Science, Vol. 3175, 145-153, Springer-Verlag, Berlin Heidelberg, New York. (2004)
15. Yang, L., Jin, R.: Distance metric learning: A comprehensive survey. [Online]. Available: http://www.cse.msu.edu/~yangliu1/frame_ survey_v2.pdf. (2006)
16. Xing, E., Ng, A., Jordan, M., Russell, S.: Distance metric learning with application to clustering with side-information. In Proceedings of the Conference on Advances in Neural Information Processing Systems, 505-512. (2003)
17. Hoi, S.C.H.,, Liu, W., Lyu, M.R., Ma, W.Y.: Learning Distance Metrics with Contextual Constraints for Image Retrieval. In Proceedings of 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2072-2078. (2006)
18. Sun, T.F., Jiang, X.H., Fu, G.L., Li, R.J., Feng, B.: Image Semantic Recognition Scheme with Semantic-binding Hierarchical Visual Vocabulary Model, In Proceedings of the 3rd International Congress on Image and Signal Processing, 1576-1582. (2010)

Xinghao Jiang received the PhD. degree in Electronic Science and Technology from Zhejiang University, Hangzhou, PR China in 2003. He is an associate professor of the School of Information Security Engineering at Shanghai Jiao Tong University, Shanghai, PR China. His current research interests include multimedia security and image retrieval, cyber information security, information hiding and watermarking, multimedia content management and rights management.

Tanfeng sun received the PhD. degree in Information and Communication Engineering from Jilin University, Jili, PR China in 2003. He is a lecturer of the School of Information Security Engineering at Shanghai Jiao Tong University, Shanghai, PR China. His current research interests include multimedia security and image retrieval, information hiding and watermarking.

Guanglei Fu received the M.S. degree from Shanghai Jiao Tong University, Shanghai, PR China in 2011. His current research interests include image and video processing, retrieval and classification, multimedia content protection.

Received: April 23, 2010; Accepted: February 20, 2011.

CIP – Каталогизација у публикацији
Народна библиотека Србије, Београд

004

COMPUTER Science and Information Systems : the International journal / Editor-in-Chief Mirjana Ivanović. – Vol. 8, No 3 (2011) - . – Novi Sad (Trg D. Obradovića 3): ComSIS Consortium, 2011 - (Belgrade : Sigra star). –30 cm

Polugodišnje. – Tekst na engleskom jeziku

ISSN 1820-0214 = Computer Science and Information Systems
COBISS.SR-ID 112261644

Cover design: V. Štavljanin
Library administration: M. Petković
Printed by: Sigra star, Beograd

ComSIS
Computer Science and Information Systems
Volume 8, Number 3, June 2011

Contents

ISSN: 1820-0214