LNCS 6786

Beniamino Murgante   Osvaldo Gervasi
Andrés Iglesias   David Taniar
Bernady O. Apduhan (Eds.)

Computational Science and Its Applications – ICCSA 2011

International Conference
Santander, Spain, June 2011
Proceedings, Part V

5 Part V

Springer

Lecture Notes in Computer Science 6786

Beniamino Murgante   Osvaldo Gervasi
Andrés Iglesias   David Taniar
Bernady O. Apduhan (Eds.)

Computational Science and Its Applications - ICCSA 2011

International Conference
Santander, Spain, June 20-23, 2011
Proceedings, Part V

Springer

Volume Editors

Beniamino Murgante
Basilicata University Potenza, Italy
E-mail: beniamino.murgante@unibas.it

Osvaldo Gervasi
University of Perugia, Italy
E-mail: osvaldo@unipg.it

Andrés Iglesias
University of Cantabria, Santander, Spain
E-mail: iglesias@unican.es

David Taniar
Monash University, Clayton, VIC, Australia
E-mail: david.taniar@infotech.monash.edu.au

Bernady O. Apduhan
Kyushu Sangyo University
Fukuoka, Japan
E-mail: bob@is.kyusan-u.ac.jp

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-21933-7 e-ISBN 978-3-642-21934-4
DOI 10.1007/978-3-642-21934-4
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011929636

CR Subject Classification (1998): C.2, H.4, F.2, H.3, D.2, C.2.4, F.1, H.5

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

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

Printed on acid-free paper

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

Preface

These multiple volumes (LNCS volumes 6782, 6783, 6784, 6785 and 6786) consist of the peer-reviewed papers from the 2011 International Conference on Computational Science and Its Applications (ICCSA 2011) held in Santander, Spain during June 20-23, 2011. ICCSA 2011 was a successful event in the International Conferences on Computational Science and Its Applications (ICCSA) conference series, previously held in Fukuoka, Japan (2010), Suwon, South Korea (2009), Perugia, Italy (2008), Kuala Lumpur, Malaysia (2007), Glasgow, UK (2006), Singapore (2005), Assisi, Italy (2004), Montreal, Canada (2003), and (as ICCS) Amsterdam, The Netherlands (2002) and San Francisco, USA (2001).

Computational science is a main pillar of most of the present research, as well as industrial and commercial activities and plays a unique role in exploiting ICT innovative technologies. The ICCSA conferences have been providing a venue to researchers and industry practitioners to discuss new ideas, to share complex problems and their solutions, and to shape new trends in computational science.

Apart from the general tracks, ICCSA 2011 also included 31 special sessions and workshops, in various areas of computational science, ranging from computational science technologies to specific areas of computational science, such as computer graphics and virtual reality. We accepted 52 papers for the general track, and 210 in special sessions and workshops. These represent an acceptance rate of 29.7%. We would like to show our appreciations to the Workshop and Special Session Chairs and co-Chairs.

The success of the ICCSA conference series, in general, and ICCSA 2011, in particular, is due to the support of many people: authors, presenters, participants, keynote speakers, Session Chairs, Organizing Committee members, student volunteers, Program Committee members, International Liaison Chairs, and people in other various roles. We would like to thank them all. We would also like to thank Springer for their continuous support in publishing ICCSA conference proceedings.

June 2011

Osvaldo Gervasi
David Taniar

Message from the ICCSA 2011 General Chairs

These five volumes contain an outstanding collection of refereed papers selected for the 11th International Conference on Computational Science and Its Applications, ICCSA 2011, held in Santander (Spain), June 20-23, 2011. We cordially invite you to visit the ICCSA website `http://www.iccsa.org` where you can find all relevant information about this interesting and exciting event.

ICCSA 2011 marked the beginning of the second decade of this conference series. Previous editions in this series of highly successful International Conferences on Computational Science and Its Applications (ICCSA) were held in Fukuoka, Japan (2010), Suwon, Korea (2009), Perugia, Italy (2008), Kuala Lumpur, Malaysia (2007), Glasgow, UK (2006), Singapore (2005), Assisi, Italy (2004), Montreal, Canada (2003), and (as ICCS) Amsterdam, The Netherlands (2002) and San Francisco, USA (2001).

As we enter the second decade of ICCSA, we realize the profound changes and spectacular advances in the world of computational science. This discipline plays a unique role in fostering new technologies and knowledge, and is crucial for most of the present research, and industrial and commercial activities. We believe that ICCSA has contributed to this change by offering a real opportunity to explore innovative approaches and techniques to solve complex problems. Reciprocally, the computational science community has enthusiastically embraced the successive editions of ICCSA, thus contributing to making ICCSA a focal meeting point for those interested in innovative, cutting-edge research about the latest and most exciting developments in the field. We are grateful to all those who have contributed to the current success of ICCSA with their continued support over the past ten years.

ICCSA 2011 would not have been made possible without the valuable contribution from many people. We would like to thank all session organizers for their diligent work, which further enhanced the conference levels and all reviewers for their expertise and generous effort which led to a very high quality event with excellent papers and presentations. We especially recognize the contribution of the Program Committee and Local Organizing Committee members for their tremendous support and for making this congress a very sucessful event.

We would like to sincerely thank our keynote speakers, who willingly accepted our invitation and shared their expertise through illuminating talks, helping us to fully meet the conference objectives.

We highly appreciate the University of Cantabria for their enthusiastic acceptance to host the conference on its main campus, their logistic assistance and additional financial support. The conference was held in the Faculty of Sciences of the University of Cantabria. We thank the Dean of the Faculty of Sciences, Ernesto Anabitarte, for his support before and during the congress, and for providing the venue of the conference and the use of all needed facilities.

ICCSA 2011 was jointly organized by the Department of Applied Mathematics and Computational Sciences and the Department of Mathematics, Statistics and Computation of the University of Cantabria, Spain. We thank both departments for their encouranging support of this conference from the very beginning. We would like to express our gratitude to the Local Organizing Committee for their persistent and enthusiastic work towards the success of this conference.

We owe special thanks to all our sponsors: the Faculty of Sciences, the University of Cantabria, the Municipality of Santander, the Regional Government of Cantabria and the Spanish Ministry of Science and Innovation, for their continuous support without which this conference would not be possible. We also thank our publisher, Springer, for their acceptance to publish the proceedings and for their kind assistance and cooperation during the editing process.

Finally, we thank all authors for their submissions and all conference attendants for making ICCSA 2011 truly an excellent forum on computational science, facilitating exchange of ideas, fostering new collaborations and shaping the future of this exciting field. Last, but certainly not least, we wish to thank our readers for their interest in these proceedings. We really hope you find in these pages interesting material and fruitful ideas for your future work.

June 2011

Andrés Iglesias
Bernady O. Apduhan

The Wisdom of Ancient Masters

In 1879, Marcelino Sanz de Sautuola and his young daugther María incidentally noticed that the ceiling of the Altamira cave was covered by images of bisons and other animals, some as old as between 25,000 and 35,000 years. They had discovered what came to be called the Sistine Chapel of Paleolithic Art. When the discovery was first made public in 1880, many experts rejected it under the belief that prehistoric man was unable to produce such beautiful and ellaborated paintings. Once their authenticity was later confirmed, it changed forever our perception of prehistoric human beings.

Today, the cave of Altamira and its paintings are a symbol of the wisdom and ability of our ancient ancestors. They remind us that our current technological development is mostly based on the work, genius and efforts of our predecessors over many generations.

The cave of Altamira (UNESCO World Heritage Site) is located in the region of Cantabria, near the city of Santander (ICCSA 2011 conference venue). The original cave is closed to the public for preservation, but conference attendees visited the "Neocave", an exact reproduction of the original space with all its cracks and textures and the permanent exhibition "The Times of Altamira", which introduces visitors to the prehistory of the peninsula and rupestrian art.

"After Altamira, all is decadence" (Pablo Picasso, famous Spanish painter)

ICCSA 2011 Welcome Message

Welcome to the proceedings of the 11th International Conference on Computational Science and Its Applications, ICCSA 2011, held in Santander, Spain.

The city of Santander is located in the self-governed region of Cantabria, on the northern coast of Spain between Asturias and the Basque Country. This beautiful region of half a million inhabitants is on the shores of the Cantabrian Sea and is crossed by a mountain range. The shores and inland valleys offer a wide variety of landscapes as a consequence of the mild, moist climate of so-called Green Spain. The coastal landscape of beaches, bays and cliffs blends together with valleys and highland areas. All along the coast there are typical traditional fishing ports and innumerable diverse beaches of soft white sand.

However, Cantabria's attractions are not limited to its natural treasures. History has provided a rich artistic and cultural heritage found in towns and villages that are outstanding in their own right. The archaeological remains and historic buildings bear the mark of a unique history starting with the world-famous Altamira cave paintings, a veritable shrine to the prehistoric age. In addition, there are remarkable remains from the Romans, the Mozarabic presence and the beginnings of the Reconquest of Spain, along with an artistic heritage of Romanesque, Gothic and Baroque styles. Examples include the Prehistoric Era (the Altamira and Puente Viesgo Caves), Roman ruins such as those of Julióbriga, medieval settlements, such as Santillana del Mar, and several examples of the civil and religious architecture of the nineteenth and twentieth centuries.

The surrounding natural landscape and the historical importance of many of its villages and buildings make this region very appealing for tourism, especially during the spring and summer seasons, when the humid, mild weather gives the region a rich and varied nature of woods and meadows. At the time of the conference, attendees enjoyed the gentle climate (with temperatures averaging 18-20 degrees Celsius) and the longest days of the year. They found themselves waiting for sunset at the beach at about 11 pm!

Capital of the autonomous region of Cantabria, the city of Santander is also a very popular destination for tourism. Based around one of the most beautiful bays in the world, this modern city is famous for its sparkling beaches of yellow sand and clean water, the hospitality of its people and the high reputation of its celebrated gastronomy, mostly based on fish and shellfish. With a population of about 200,000 inhabitants, Santander is a very safe city, with a vibrant tourist scene filled with entertainment and a high quality of life, matching the best standards in the world. The coastal side of the city boasts a long string of top–quality beaches and recreational areas, such as the Magdalena Peninsula, the Sardinero and Mataleñas Park. There are several beaches and harbors limiting the city on the northern side, toward the southern part there is the old city center and a bit further on the green mountains. We proudly say that Santander is between the blue and the green.

The University of Cantabria (in Spanish, *the Universidad de Cantabria, UC*) is the only public university in Cantabria, Spain. It was founded in 1972 and is organized in 12 faculties and schools. With about 13,000 students and 1,000 academic staff, the University of Cantabria is one of the most reputed universities in the country, ranking in the highest positions of Spanish universities in relation to its size. Not surprisingly, it was selected as a Campus of International Excellence by the Spanish Government in 2009.

Besides the technical sessions and presentations, ICCSA 2011 provided an interesting, must-attend social program. It started with a Welcome Reception at the Royal Palace of the Magdalena (Sunday 19), the most emblematic building of Santander and also the most visited place in the city. The royal family used the palace during the period 1913–1930 as a base for numerous recreational and sporting activities, and the king sometimes also held government meetings at the property. Conference delegates had the wonderful opportunity to visit this splendid palace, enjoy the magnificent views and see some rooms where royalty lived. The Gala Dinner (Tuesday 21) took place at the Grand Casino, in the "Sardinero" area, a regal, 1920's building with large windows and spacious terraces offering superb views of the Sardinero beach. The Casino was King Alfonso XIII and Queen Victoria Eugenia's main place of entertainment during their summer holidays in the city between 1913 and 1930. The gala also included some cultural and musical events. Finally, a half-day conference tour (Wednesday 22) covered the "live museum" of the Middle Ages, Santillana del Mar (a medieval town with cobbled streets, declared "Site of Artistic and Historical Importance" and one of the best-known cultural and tourist centers in Cantabria) and the Altamira Neocave, an exact reproduction of the original Altamira cave (now closed to the public for preservation) with all its cracks and textures and the permanent exhibition "The Times of Altamira", which introduces visitors to the prehistory of the peninsula and rupestrian art.

To close the conference, attendees could join the people of Santander for St. John's day, celebrated in the night between June 23 and 24 to commemorate the summer solstice with bonfires on the beach.

We believe that all these attractions made the conference an unforgettable experience.

On behalf of the Local Organizing Committee members, I thank all attendees fot their visit.

June 2011 Andrés Iglesias

Message from the Chairs of the Session: 6th International Workshop on "Geographical Analysis, Urban Modeling, Spatial Statistics" (GEOG-AN-MOD 2011)

During the past few decades the main problem in geographical analysis was the lack of spatial data availability. Nowadays the wide diffusion of electronic devices containing geo-referenced information generates a great production of spatial data. Volunteered geographic information activities (e.g., Wikimapia, OpenStreetMap), public initiatives (e.g., spatial data infrastructures, geo-portals) and private projects (e.g., Google Earth, Microsoft Virtual Earth, etc.) produced an overabundance of spatial data, which, in many cases, do not help the efficiency of decision processes. The increase of geographical data availability has not been fully coupled by an increase of knowledge to support spatial decisions.

The inclusion of spatial simulation techniques in recent GIS software favored the diffusion of these methods, but in several cases led to mechanisms based on which buttons have to be pressed without having geography or processes in mind. Spatial modeling, analytical techniques and geographical analyses are therefore required in order to analyze data and to facilitate the decision process at all levels, with a clear identification of the geographical information needed and reference scale to adopt. Old geographical issues can find an answer thanks to new methods and instruments, while new issues are developing, challenging researchers for new solutions. This workshop aims at contributing to the development of new techniques and methods to improve the process of knowledge acquisition.

Conference themes include:

Geostatistics and spatial simulation
Agent-based spatial modeling
Cellular automata spatial modeling
Spatial statistical models
Space-temporal modeling
Environmental modeling
Geovisual analytics, geovisualization, visual exploratory data analysis
Visualization and modeling of track data
Spatial optimization
Interaction simulation models
Data mining, spatial data mining
Spatial data warehouse and spatial OLAP
Integration of spatial OLAP and spatial data mining
Spatial decision support systems

Spatial multicriteria decision analysis
Spatial rough set
Spatial extension of fuzzy set theory
Ontologies for spatial analysis
Urban modeling
Applied geography
Spatial data analysis
Dynamic modeling
Simulation, space-time dynamics, visualization and virtual reality.

Giuseppe Borruso
Beniamino Murgante
Stefania Bertazzon

Message from the Chairs of the Session: "Cities, Technologies and Planning" (CTP 2011)

'Share' term has turned into a key issue of many successful initiatives in recent times. Following the advent of Web 2.0, positive experiences based on mass collaboration generated "Wikinomics" hnd ave become 'Socialnomics", where 'Citizens are voluntary sensors'.

During the past few decades, the main issue in GIS implementation has been the availability of sound spatial information. Nowadays, the wide diffusion of electronic devices providing geo-referenced information resulted in the production of extensive spatial information datasets. This trend has led to "GIS wikification", where mass collaboration plays a key role in the main components of spatial information frameworks (hardware, software, data, and people).

Some authors (Goodchild, 2007) talk about 'volunteered geographic information' (VGI), as the harnessing of tools to create, assemble, and disseminate geographic information provided by individuals voluntarily creating their own contents by marking the locations of occurred events or by labeling certain existing features not already shown on a map. The term "neogeography" is often adopted to describe peoples activities when using and creating their own maps, geo-tagging pictures, movies, websites, etc. It could be defined as a new bottom up approach to geography prompted by users, therefore introducing changes in the roles of traditional' geographers and consumers' of geographical contents themselves. The volunteered approach has been adopted by important American organizations, such as US Geological Survey, US Census Bureau, etc. While technologies (e.g. GPS, remote sensing, etc.) can be useful in producing new spatial data, volunteered activities are the only way to update and describe such data. If spatial data have been produced in various ways, remote sensing, sensor networks and other electronic devices generate a great flow of relevant spatial information concerning several aspects of human activities or of environmental phenomena monitoring. This 'information-explosion era' is characterized by a large amount of information produced both by human activities and by automated systems; the capturing and the manipulation of this information leads to 'urban computing' and represents a sort of bridge between computers and the real world, accounting for the social dimension of human environments. This technological evolution produced a new paradigm of urban development, called 'u-City'. Such phenomena offer new challenges to scholars (geographers, engineers, planners, economists, sociologists, etc.) as well as to spatial planners in addressing spatial issues and a wealth of brand-new, updated data, generally created by people who are interested in geographically related phenomena. As attention is to date dedicated to visualization and content creation, little has been done from the spatial analytical point of view and in involving users as citizens in participatory geographical activities.

Conference themes include:

SDI and planning
Planning 2.0, participation 2.0
Urban social networks, urban sensing
E-democracy, e-participation, participatory GIS
Technologies for e-participation, policy modeling, simulation and visualization
Second Life and participatory games
Ubiquitous computing environment; urban computing; ubiquitous-city
Neogeography
Collaborative mapping
Geotagging
Volunteered geographic information
Crowdsourcing
Ontologies for urban planning
City Gml
Geo-applications for mobile phones
Web 2.0, Web 3.0
Wikinomics, socialnomics
WikiCities
Maps mash up
Tangible maps and planning
Augmented reality,
Complexity assessment and mapping

Giuseppe Borruso
Beniamino Murgante

Message from the Chairs of the Session: 11^{th} Annual International Workshop on "Computational Geometry and Applications" (CGA 2011)

The 11th International Workshop on Computational Geometry and Applications CGA 2011, held in conjunction with the International Conference on Computational Science and Applications, took place in Santander, Spain. The workshop has run annually since it was founded in 2001, and is intended as an international forum for researchers in computational geometry and related areas, with the goal of advancing the state of research in computational geometry and related disciplines. This year, the workshop was chaired for 11th year by CGA workshop series Founding Chair Marina Gavrilova, University of Calgary, joined by co-Chair Ovidiu Daescu, University of Texas at Dallas. Selected papers from the previous CGA Workshops have appeared in special issues in the following highly regarded journals: *International Journal of Computational Geometry and Applications*, Springer (three special issues), *International Journal of Computational Science and Engineering* (IJCSE), Journal of CAD/CAM , *Transactions on Computational Sciences*, Springer. A special issue comprising best papers presented at CGA 2011 is currently being planned.

The workshop attracts international attention and receives papers presenting high-quality original research in the following tracks:

- Theoretical computational geometry
- Applied computational geometry
- Optimization and performance issues in geometric algorithms implementation Workshop topics of interest include:
- Design and analysis of geometric algorithms
- Geometric algorithms in path planning and robotics
- Computational geometry in biometrics
- Intelligent geometric computing
- Geometric algorithms in computer graphics and computer vision
- Voronoi diagrams and their generalizations
- 3D Geometric modeling
- Geometric algorithms in geographical information systems
- Algebraic geometry
- Discrete and combinatorial geometry
- Implementation issues and numerical precision
- Applications in computational biology, physics, chemistry, geography, medicine, education
- Visualization of geometric algorithms

CGA 2011 was located in beautiful Santander, Cantabria, Spain. Santander, the capital city of Cantabria, is located on the northern coast of Spain, between Asturias and the Basque Country overlooking the Cantabrian Sea, and is surrounded by beaches. The conference preceded the Spanish Meeting on Computational Geometry, which took place in Madrid, facilitating interested researchers to attend both events. The 14 articles presented in this Springer LNCS proceeding volume represent papers selected fromi a large number of submissions to this year's workshop. We would like to express our sincere gratitude to the following International Program Committee members who performed their duties diligently and provided constructive feedback for authors to improve on their presentation:

Tetsuo Asano (Japan Advanced Institute of Science and Technology, Japan)
Sergei Bereg (University of Texas at Dallas, USA)
Karoly Bezdek (University of Calgary, Canada)
Ovidiu Daescu (University of Texas at Dallas, USA)
Mirela Damian (Villanova University, USA)
Tamal Dey (Ohio State University, USA)
Marina L. Gavrilova (University of Calgary, Canada)
Christopher Gold (University of Glamorgan, UK)
Hisamoto Hiyoshi (Gunma University, Japan)
Andrés Iglesias (University of Cantabria, Spain)
Anastasia Kurdia (Smith College, USA)
Deok-Soo Kim (Hanyang University, Korea)
Ivana Kolingerova (Unversity of West Bohemia, Czech Republic)
Nikolai Medvedev (Novosibirsk Russian Academy of Science, Russia)
Asish Mukhopadhyay (University of Windsor, Canada)
Dimitri Plemenos (Université de Limoges, France)
Val Pinciu (Southern Connecticut State University, USA)
Jon Rokne (University of Calgary, Canada)
Carlos Seara (Universitat Politecnica de Catalunya, Spain)
Kokichi Sugihara (University of Tokyo, Japan)
Vaclav Skala (University of West Bohemia, Czech Republic)
Muhammad Sarfraz (KFUPM, Saudi Arabia)
Alexei Sourin (Nanyang Technological University, Singapore)
Ryuhei Uehara (Japan Advanced Institute of Science and Technology, Japan)
Chee Yap (New York University, USA)
Kira Vyatkina (Sanct Petersburg State University, Russia)

We also would like to acknowledge the independent referees, ICCSA 2011 organizers, sponsors, volunteers, and Springer for their continuing collaboration and support.

Marina C. Gavrilova
Ovidiu Daescu

Message from the Chair of the Session: 3^{rd} International Workshop on "Software Engineering Processes and Applications" (SEPA 2011)

The Third International Workshop on Software Engineering Processes and Applications (SEPA 2011) covered the latest developments in processes and applications of software engineering. SEPA includes process models, agile development, software engineering practices, requirements, system and design engineering including architectural design, component level design, formal methods, software modeling, testing strategies and tactics, process and product metrics, Web engineering, project management, risk management, and configuration management and all those areas which are related to the processes and any type of software applications. This workshop attracted papers from leading researchers in the field of software engineering and its application areas. Seven regular research papers were accepted as follows.

Sanjay Misra, Ibrahim Akman and Ferid Cafer presented a paper on "A Multi-Paradigm Complexity Metric(MCM)" The authors argued that there are not metrics in the literature for multi-paradigm. MCM is developed by using function points and procedural and object–oriented language's features. In this view, MCM involves most of the factors which are responsible for the complexity of any multi-paradigm language. MCM can be used for most programming paradigms, including both procedural and object–oriented languages.

Mohamed A. El-Zawawy's paper entitled 'Flow Sensitive-Insensitive Pointer Analysis Based Memory Safety for Multithreaded Programs' presented approaches for the pointer analysis and memory safety of multithreaded programs as simply structured type systems. The author explained that in order to balance accuracy and scalability, the proposed type system for pointer analysis of multithreaded programs is flow-sensitive, which invokes another flow-insensitive type system for parallel constructs.

Cesar Pardo, Francisco Pino, Felix Garcia, Francisco Romero, Mario Piattini, and Maria Teresa Baldassarre presented their paper entitled 'HProcessTOOL: A Support Tool in the Harmonization of Multiple Reference Models'. The authors have developed the tool HProcessTOOL, which guides harmonization projects by supporting specific techniques, and supports their management by controlling and monitoring the resulting harmonization projects. The validation of the tool is performed by two case studies.

Wasi Haider Butt, Sameera Amjad and Farooque Azam presented a paper on 'Requirement Conflicts Resolution: Using Requirement Filtering and Analysis'. The authors presented a systematic approach toward resolving software requirements spanning from requirement elicitation to the requirement analysis activity of the requirement engineering process. The authors developed a model 'conflict resolution strategy' (CRS) which employs a requirement filter and an analysis strategy for resolving any conflict arising during software development. They also implemented their model on a real project.

Rajesh Prasad, Suneeta Agarwal, Anuj Kumar Sharma, Alok Singh and Sanjay Misra presented a paper on 'Efficient Algorithm for Detecting Parameterized Multiple Clones in a Large Software System'. In this paper the authors have tried to solve the word length problem in a bit-parallel parameterized matching by extending the BLIM algorithm of exact string matching. The authors further argued that the extended algorithm is also suitable for searching multiple patterns simultaneously. The authors presented a comparison in support of their algorithm.

Takahiro Uchiya and Tetsuo Kinoshita presented the paper entitled 'Behavior Analyzer for Developing Multiagent Systems on Repository-Based Multiagent Framework'. In this paper the authors proposed an interactive design environment of agent system (IDEA) founded on an agent-repository-based multiagent framework. They focused on the function of the behavior analyzer for developing multiagent systems and showed the effectiveness of the function.

Jose Alfonso Aguilar, Irene Garrigos, and Jose-Norberto Mazon presented a paper on 'Impact Analysis of Goal-Oriented Requirements in Web Engineering'. This paper argues that Web developers need to know dependencies among requirements to ensure that Web applications finally satisfy the audience. The authors developed an algorithm to deal with dependencies among functional and non-functional requirements so as to understand the impact of making changes when developing a Web application.

Sanjay Misra

Message from the Chair of the Session: 2^{nd} International Workshop on "Software Quality" (SQ 2011)

Following the success of SQ 2009, the Second International Workshop on "Software Quality" (SQ 2011) was organized in conjunction with ICCSA 2011. This workshop extends the discussion on software quality issues in the modern software development processes. It covers all the aspects of process and product quality, quality assurance and standards, quality planning, quality control and software quality challenges. It also covers the frontier issues and trends for achieving the quality objectives. In fact this workshop covers all areas, that are concerned with the quality issue of software product and process. In this workshop, we featured nine articles devoted to different aspects of software quality.

Roberto Espinosa, Jose Zubcoff, and Jose-Norberto Mazon's paper entitled "A Set of Experiments to Consider Data Quality Criteria in Classification Techniques for Data Mining" analyzed data–mining techniques to know the behavior of different data quality criteria from the sources. The authors have conducted a set of experiments to assess three data quality criteria: completeness, correlation and balance of data.

In their paper, Ivaylo Spassov, Valentin Pavlov, Dessislava Petrova-Antonova, and Sylvia Ilieva's have developed a tool "DDAT: Data Dependency Analysis Tool for Web Service Business Processes". The authors have implemented and shown experimental results from the execution of the DDAT over BPEL processes.

Filip Radulovic and Raul Garca-Castro presented a paper on "Towards a Quality Model for Semantic Technologies". The authors presented some well-known software quality models, after which a quality model for semantic technologies is designed by extending the ISO 9126 quality model.

Luis Fernandez-Sanz and Sanjay Misra authored the paper "Influence of Human Factors in Software Quality and Productivity". The authors first analyzed the existing contributions in the area and then presented empirical data from specific initiatives to know more about real practices and situations in software organizations.

Eudisley Anjos, and Mario Zenha-Rela presented a paper on "A Framework for Classifying and Comparing Software Architecture Tools for Quality Evaluation". This framework identifies the most relevant features for categorizing different architecture evaluation tools according to six different dimensions. The authors reported that the attributes that a comprehensive tool should support include: the ability to handle multiple modeling approaches, integration with the industry standard UML or specific ADL, support for trade–off analysis of competing quality attributes and the reuse of knowledge through the build-up of new architectural patterns.

Hendrik Decker presented a paper on “Causes of the Violation of Integrity Constraints for Supporting the Quality of Databases”. He presented a quality metric with the potential of more accuracy by measuring the causes. He further argued that such measures also serve for controlling quality impairment across updates.

Csaba Nagy, Laszlo Vidacs , Rudolf Ferenc, Tibor Gyimothy Ferenc Kocsis, and Istvan Kovacs’s presented a paper on “Complexity measures in a 4GL environment”. The authors discussed the challenges in adopting the metrics from 3GL environments. Based on this, they presented a complexity measure in 4GL environments. They performed the experimentations and demonstrated the results.

Lukasz Radlinski’s paper on “A Framework for Integrated Software Quality Prediction Using Bayesian Nets” developed a framework for integrated software quality prediction. His framework is developed and formulated using a Bayesian net, a technique that has already been used in various software engineering studies. The author argues that his model may be used in decision support for software analysts and managers.

Seunghun Park, Sangyoon Min, and Doohwan Bae authored the paper entitled “Process Instance Management Facilities Based on the Meta-Process Models”. Based on the metar-process models, the authors proposed a process model and two types of process instance models: the structural instance model and the behavioral instance model. The authors’ approach enables a project manager to analyze structural and behavioral properties of a process instance and allows a project manager to make use of the formalism for management facilities without knowledge of the formalism.

Sanjay Misra

Message from the Chairs of the Session: "Remote sensing Data Analysis, Modeling, Interpretation and Applications: From a Global View to a Local Analysis" (RS 2011)

Remotely sensed data provide temporal and spatial consistent measurements useful for deriving information on the dynamic nature of Earth surface processes (sea, ice, land, atmosphere), detecting and identifying land changes, discovering cultural resources, studying the dynamics of urban expansions. Thanks to the establishment and maintenance of long-term observation programs, presently a huge amount of multiscale and multifrequency remotely sensed data are available.

To fully exploit such data source for various fields of application (environmental, cultural heritage, urban analysis, disaster management) effective and reliable data processing, modeling and interpretation are required. This session brought together scientists and managers from the fields of remote sensing, ICT, geospatial analysis and modeling, to share information on the latest advances in remote sensing data analysis, product development, validation and data assimilation.

Main topics included:

Remotely sensed data – Multispectral satellite : from medium to very high spatial resolution; airborne and spaceborne Hypespectral data; open data source (Modis, Vegetation, etc..); airborne Laser Scanning; airborne and spaceborne Radar imaging; thermal imaging; declassified Intelligence Satellite Photographs (Corona, KVR); ground remote sensing

Methods and procedures – change detection; classification Data fusion / Data integration; data mining; geostatistics and Spatial statistics; image processing; image interpretation; linear and on linear statistical analysis; segmentation Pattern recognition and edge detection; time space modeling

Fields of application and products – archaeological site discovery; cultural Heritage management; disaster management; environmental sciences; mapping Landscape and digital elevation models; land cover analysis; open source softwares; palaeoenvironmental studies; time series

Nicola Masini
Rosa Lasaponara

Message from the Chairs of the Session: "Approximation, Optimization and Applications" (AOA 2011)

The objective of the session Approximation, Optimization and Applications during the 11th International Conference on Computational Science and Its Applications was to bring together scientists working in the areas of Approximation Theory and Numerical Optimization, including their applications in science and engineering.

Hypercomplex function theory, renamed Clifford analysis in the 1980s, studies functions with values in a non-commutative Clifford algebra. It has its roots in quaternionic analysis, developed as another generalization of the classic theory of functions of one complex variable compared with the theory of functions of several complex variables. It is well known that the use of quaternions and their applications in sciences and engineering is increasing, due to their advantages for fast calculations in 3D and for modeling mathematical problems. In particular, quasi-conformal 3D-mappings can be realized by regular (monogenic) quaternionic functions. In recent years the generalization of classical polynomials of a real or complex variable by using hypercomplex function theoretic tools has been the focus of increased attention leading to new and interesting problems. All these aspects led to the emergence of new software tools in the context of quaternionic or, more generally, Clifford analysis.

Irene Falcão
Ana Maria A.C. Rocha

Message from the Chair of the Session: "Symbolic Computing for Dynamic Geometry" (SCDG 2011)

The papers comprising in the Symbolic Computing for Dynamic Geometry technical session correspond to talks delivered at the conference. After the evaluation process, six papers were accepted for oral presentation, according to the recommendations of the reviewers. Two papers, "Equal bisectors at a vertex of a triangle" and "On Equivalence of Conditions for a Quadrilateral to Be Cyclica", study geometric problem by means of symbolic approaches.

Another contributions deal with teaching ("Teaching geometry with TutorMates" and "Using Free Open Source Software for Intelligent Geometric Computing"), while the remaining ones propose a framework for the symbolic treatment of dynamic geometry ("On the Parametric Representation of Dynamic Geometry Constructions") and a formal library for plane geometry ("A Coq-based Library for Interactive and Automated Theorem Proving in Plane Geometry").

Francisco Botana

Message from the Chairs of the Session: "Computational Design for Technology Enhanced Learning" (CD4TEL 2011)

Providing computational design support for orchestration of activities, roles, resources, and systems in technology-enhanced learning (TEL) is a complex task. It requires integrated thinking and interweaving of state-of-the-art knowledge in computer science, human–computer interaction, pedagogy, instructional design and curricular subject domains. Consequently, even where examples of successful practice or even standards and specifications like IMS learning design exist, it is often hard to apply and (re)use these efficiently and systematically. This interdisciplinary technical session brought together practitioners and researchers from diverse backgrounds such as computer science, education, and cognitive sciences to share their proposals and findings related to the computational design of activities, resources and systems for TEL applications.

The call for papers attracted 16 high-quality submissions. Each submission was reviewed by three experts. Eventually, five papers were accepted for presentation. These contributions demonstrate different perspectives of research in the CD4TEL area, dealing with standardization in the design of game-based learning; the integration of individual and collaborative electronic portfolios; the provision of an editing environment for different actors designing professional training; a simplified graphical notation for modeling the flow of activities in IMS learning design units of learning; and a pattern ontology-based model to support the selection of good-practice scripts for designing computer–supported collaborative learning.

Michael Derntl
Manuel Caeiro-Rodríguez
Davinia Hernández-Leo

Message from the Chair of the Session: "Chemistry and Materials Sciences and Technologies" (CMST 2011)

The CMST workshop is a typical example of how chemistry and computer science benefit from mutual interaction when operating within a grid e-science environment. The scientific contributions to the workshop, in fact, contain clear examples of chemical problems solved by exploiting the extra power offered by the grid infrastructure to the computational needs of molecular scientists when trying to push ahead the frontier of research and innovation.

Ideal examples of this are the papers on the coulomb potential decomposition in the multiconfiguration time–dependent Hartree method, on the extension of the grid–empowered simulator GEMS to the a priori evaluation of the crossed beam measurements and on the evaluation of the concentration of pollutants when using a box model version of the Community Multiscale Air Quality Modeling System 4.7. Another example of such progress in computational molecular science is offered by the paper illustrating the utilization of a fault–tolerant workflow for the DL-POLY package for molecular dynamics studies.

At the same time molecular science studies are an excellent opportunity for investigating the use of new (single or clustered) GPU chips as in the case of the papers related to their use for computationally demanding quantum calculations of atom diatom reactive scattering. In addition, of particular interest are the efforts spent to develop tools for evaluating user and service quality to the end of promoting collaborative work within virtual organizations and research communities through the awarding and the redeeming of credits.

Antonio Laganà

Message from the Chairs of the Session: "Cloud for High Performance Computing" (C4HPC 2011)

On behalf of the Program Committee, it is a pleasure for us to introduce the proceedings of this First International Workshop on Cloud for High–Performance Computing held in Santander (Spain) in 2011 during the 11th International Conference on Computational Science and Its Applications. The conference joined high quality researchers around the world to present the latest results in the usage of cloud computing for high–performance computing.

High–performance computing, or HPC, is a great tool for the advancement of science, technology and industry. It intensively uses computing resources, both CPU and storage, to solve technical or scientific problems in minimum time. It also uses the most advanced techniques to achieve this objective and evolves along with computing technology as fast as possible. During the last few years we have seen the introduction of new hardware isuch as multi-core and GPU representing a formidable challenge for the scientific and technical developers that need time to absorb these additional characteristics. At the same time, scientists and technicians have learnt to make faster and more accurate measurements, accumulating a large set of data which need more processing capacity. While these new paradigms were entering the field of HPC, virtualization was suddenly introduced in the market, generating a new model for provisioning computing capacity: the cloud. Although conceptually the cloud is not completely new, because it follows the old dream of computing as a utility, it has introduced new characteristics such as elasticity, but at the cost of losing some performance.

Consequently, HPC has a new challenge: how to tackle or solve this reduction in performance while adapting methods to the elasticity of the new platform. The initial results show the feasibility of using cloud infrastructures to execute HPC applications. However, there is also some consensus that the cloud is not the solution for grand challenges, which will still require dedicated supercomputers. Although recently a cluster of more than 4000 CPUs has been deployed, there are still many technical barriers to allow technicians to use it frequently. This is the reason for this workshop which we had the pleasure of introducing.

This First International Workshop on Cloud for High–Performance Computing was an original idea of Osvaldo Gervasi. We were working on the proposal of a COST action devoted to the cloud for HPC which would link the main researchers in Europe. He realized that the technical challenges HPC has to solve in the next few years to use the Cloud efficiently, need the collaboration of as many scientists and technicians as possible as well as to rethink the way the applications are executed.

This first workshop, which deserves in the next ICCSA conferences, joined together experts in the field that presented high quality research results in the area. They include the first approximations of topology methods such as cellular data system to cloud to be used to process data. Data are also the main issue for the TimeCloud front end, an interface for time series analysis based on Hadop and Hbase, designed to work with massive datasets. In fact, cloud can generate such a large amount of data when accurate information about its infrastructure and executing applications is needed. This is the topic of the third paper which introduce LISA algorithm to tackle the problem of information retrieval in cloud environment where the methods must adapt to the elasticity, scalability and possibility of failure. In fact, to understand Cloud infrastructures, researchers and technicians will need these series of data as well as the usage of tools that allow better knowledge to be gained. In this sense, iCanCloud, a novel simulator of cloud infrastructures, is introduced presenting its results for the most used and cited service: Amazon.

We strongly believe that the reader will enjoy the selected papers, which represent only a minimal, but important, part of the effervescent activity in Cloud for HPC. This selection was only possible thanks to the members of the Program Committee, all of them supporting actively the initiative. We appreciate their commitment to the workshop. Also, we want to thank all of the reviewers who kindly participated in the review of the papers and, finally, to all the scientists who submitted a paper, even if it was not accepted. We hope that they will have the opportunity to join us in the next editions.

Andrés Gomez
Osvaldo Gervasi

ICCSA 2011 Invited Speakers

Ajith Abraham
Machine Intelligence Research Labs, USA

Marina L. Gavrilova
University of Calgary, Canada

Yee Leung
The Chinese University of Hong Kong, China

Evolving Future Information Systems: Challenges, Perspectives and Applications

Ajith Abraham

Machine Intelligence Research Labs, USA
ajith.abraham@ieee.org

Abstract

We are blessed with the sophisticated technological artifacts that are enriching our daily lives and society. It is believed that the future Internet is going to provide us with the framework to integrate, control or operate virtually any device, appliance, monitoring systems, infrastructures etc. The challenge is to design intelligent machines and networks that could communicate and adapt according to the environment. In this talk, we first present the concept of a digital ecosystem and various research challenges from several application perspectives. Finally, we present some real–world applications.

Biography

Ajith Abraham received a PhD degree in Computer Science from Monash University, Melbourne, Australia. He is currently the Director of Machine Intelligence Research Labs (MIR Labs), Scientific Network for Innovation and Research Excellence, USA, which has members from more than 75 countries. He serves/has served the editorial board of over 50 international journals and has also guest edited 40 special issues on various topics. He has authored/co-authored more than 700 publications, and some of the works have also won best paper awards at international conferences. His research and development experience includes more than 20 years in industry and academia. He works in a multidisciplinary environment involving machine intelligence, network security, various aspects of networks, e-commerce, Web intelligence, Web services, computational grids, data mining, and their applications to various real-world problems. He has given more than 50 plenary lectures and conference tutorials in these areas.

Dr. Abraham is the Chair of IEEE Systems Man and Cybernetics Society Technical Committee on Soft Computing. He is a Senior Member of the IEEE, the IEEE Computer Society, the Institution of Engineering and Technology (UK) and the Institution of Engineers Australia (Australia). He is actively involved in the Hybrid Intelligent Systems (HIS), Intelligent Systems Design and Applications (ISDA), Information Assurance and Security (IAS), and Next–Generation Web Services Practices (NWeSP) series of international conferences, in addition to other conferences. More information can be found at: `http://www.softcomputing.net`.

Recent Advances and Trends in Biometric

Marina L. Gavrilova

Department of Computer Science, University of Calgary
marina@cpsc.ucalgary.ca

Extended Abstract

The area of biometric, without a doubt, is one of the most dynamic areas of interest, which recently has displayed a gamut of broader links to other fields of sciences. Among those are visualization, robotics, multi-dimensional data analysis, artificial intelligence, computational geometry, computer graphics, e-learning, data fusion and data synthesis. The theme of this keynote is reviewing the state of the art in multi-modal data fusion, fuzzy logic and neural networks and its recent connections to advanced biometric research.

Over the past decade, multimodal biometric systems emerged as a feasible and practical solution to counterweight the numerous disadvantages of single biometric systems. Active research intoi the design of a multimodal biometric system has started, mainly centered around: types of biometrics, types of data acquisition and decision-making processes. Many challenges originating from non-uniformity of biometric sources and biometric acquisition devices result in significant differences on which information is extracted, how is it correlated, the degree of allowable error, cost implications, ease of data manipulation and management, and also reliability of the decisions being made. With the additional demand of computational power and compact storage, more emphasis is shifted toward database design and computational algorithms.

One of the actively researched areas in multimodal biometric systems is information fusion. Which information needs to be fused and what level is needed to obtain the maximum recognition performance is the main focus of current research. In this talk I concentrate on an overview of the current trends in recent multimodal biometric fusion research and illustrate in detail one fusion strategy: rank level fusion. From the discussion, it is seen that rank level fusion often outperforms other methods, especially combined with powerful decision models such as Markov chain or fuzzy logic.

Another aspect of multi-modal biometric system development based on neural networks is discussed further. Neural networks have the capacity to simulate learning processes of a human brain and to analyze and compare complex patters, which can originate from either single or multiple biometric sources, with amazing precision. Speed and complexity have been the downsides of neural networks, however, recent advancements in the area, especially in chaotic neural networks, allow these barriers to be overcome.

The final part of the presentation concentrates on emerging areas utilizing the above developments, such as decision making in visualization, graphics, e-learning, navigation, robotics, and security of web-based and virtual worlds. The extent to which biometric advancements have an impact on these emerging areas makes a compelling case for the bright future of this area.

References

1. Ross, A., Nandakumar, K., and Jain, A.K., Handbook of multibiometrics, New York, Springer (2006).
2. Jain, A.K., Ross, A., Prabhakar, S., An introduction to biometric recognition, IEEE Trans. on Circuits and Systems for Video Technology, Special Issue on Image- and Video-Based Biometrics, 14 (1): 420 (2004)
3. Nandakumar, K., Jain, A.K., Ross, A., Fusion in multibiometric identification systems: What about the missing data?, in LNCS 5558: 743752, Springer (2009).
4. Monwar, M. M., and Gavrilova, M.L., A multimodal biometric system using rank level fusion approach, IEEE Trans. SMC - B: Cybernetics, 39(4): 867-878 (2009).
5. Monwar, M. M., and Gavrilova, M.L., Secured access control through Markov chain based rank level fusion method, in proc. of 5th Int. Conf. on Computer Vision Theory and Applications (VISAPP), 458-463, Angres, France (2010).
6. Monwar, M. M., and Gavrilova, M.L., FES: A system of combining face, ear and signature biometrics using rank level fusion, in proc. 5th IEEE Int. Conf. IT: New Generations, pp 922-927, (2008).
7. Wang, C., Gavrilova, M.L., Delaunay Triangulation Algorithm for Fingerprint Matching. ISVD'2006. pp.208 216
8. Wecker, L., Samavati, F.F., Gavrilova, M.L., Iris synthesis: a reverse subdivision application. GRAPHITE'2005. pp.121 125
9. Anikeenko, A.V., Gavrilova, M.L., Medvedev, N.N., A Novel Delaunay Simplex Technique for Detection of Crystalline Nuclei in Dense Packings of Spheres. ICCSA (1)'2005. pp.816 826
10. Luchnikov, V.A., Gavrilova, M.L., Medvedev, N.N., Voloshin, V. P., The Voronoi-Delaunay approach for the free volume analysis of a packing of balls in a cylindrical container. Future Generation Comp. Syst., 2002: 673 679
11. Frischholz, R., and Dieckmann, U., BioID: A multimodal biometric identification system, IEEE Computer, 33 (2): 64-68 (2000).
12. Latifi, S., Solayappan, N. A survey of unimodal biometric methods, in proc. of Int. Conf. on Security & Management, 57-63, Las Vegas, USA (2006).
13. Dunstone, T., and Yager, N., Biometric system and data analysis: Design, evaluation, and data mining. Springer, New York (2009).
14. Ho, T.K., Hull, J.J., and Srihari, S.N., Decision combination in multiple classifier systems, IEEE Trans. on Pattern Analysis and Machine Intelligence, 16 (1): 66-75 (1994)

Biography

Marina L. Gavrilova is an Associate Professor in the Department of Computer Science, University of Calgary. Prof. Gavrilova's research interests lie in the area of computational geometry, image processing, optimization, spatial and biometric modeling. Prof. Gavrilova is founder and co-director of two innovative research laboratories: the Biometric Technologies Laboratory: Modeling and Simulation and the SPARCS Laboratory for Spatial Analysis in Computational Sciences. Prof. Gavrilova publication list includes over 120 journal and conference papers, edited special issues, books and book chapters, including World Scientific Bestseller of the Month (2007) *Image Pattern Recognition: Synthesis and Analysis in Biometric* and the Springer book Computational Intelligence: A Geometry-Based Approach. Together with Dr. Kenneth Tan, Prof. Gavrilova founded the ICCSA series of successful international events in 2001. She founded and chaired the International Workshop on Computational Geometry and Applications for over ten years, was co-Chair of the International Workshop on Biometric Technologies BT 2004, Calgary, served as Overall Chair of the Third International Conference on Voronoi Diagrams in Science and Engineering (ISVD) in 2006, was Organizing Chair of WADS 2009 (Banff), and general chair of the International Conference on Cyberworlds CW2011 (October 4-6, Banff, Canada). Prof. Gavrilova is an Editor-in-Chief of the successful LNCS Transactions on Computational Science Journal, Springer-Verlag since 2007 and serves on the Editorial Board of the International Journal of Computational Sciences and Engineering, CAD/CAM Journal and Journal of Biometrics. She has been honored with awards and designations for her achievements and was profiled in numerous newspaper and TV interviews, most recently being chosen together with other outstanding Canadian scientists to be featured in the National Museum of Civilization and National Film Canada production.

Theories and Applications of Spatial-Temporal Data Mining and Knowledge Discovery

Yee Leung

The Chinese University of Hong Kong, China
yeeleung@cuhk.edu.hk

Abstract

Basic theories of knowledge discovery in spatial and temporal data are examined in this talk. Fundamental issues in the discovery of spatial structures and processes will first be discussed. Real-life spatial data mining problems are employed as the background on which concepts, theories and methods are scrutinized. The unraveling of land covers, seismic activities, air pollution episodes, rainfall regimes, epidemics, patterns and concepts hidden in spatial and temporal data are employed as examples to illustrate the theoretical arguments and algorithms performances. To round up the discussion, directions for future research are outlined.

Biography

Yee Leung is currently Professor of Geography and Resource Management at The Chinese University of Hong Kong. He is also the Associate Academic Director of the Institute of Space and Earth Information Science of The Chinese University of Hong Kong. He is adjunct professor of several universities in P.R. China. Professor Leung had also served on public bodies including the Town Planning Board and the Environmental Pollution Advisory Committee of Hong Kong SAR. He is now Chair of The Commission on Modeling Geographical Systems, International Geographical Union, and Chair of The Commission on Quantitative and Computational Geography of The Chinese Geographical Society. He serves on the editorial board of several international journals such as *Annals of Association of American Geographers*, *Geographical Analysis*, *GeoInformatica*, *Journal of Geographical Systems*, *Acta Geographica Sinica*, *Review of Urban and Regional Development Studies*, etc. Professor Leung is also Council member of The Society of Chinese Geographers.

Professor Leung carried out pioneer and influential research in imprecision/uncertainty analysis in geography, intelligent spatial decision support systems, geocomputation (particularly on fuzzy sets, rough sets, spatial statistics, fractal analysis, neural networks and genetic algorithms), and knowledge discovery and data mining. He has obtained more than 30 research grants, authored and co-authored six books and over 160 papers in international journals and book chapters on geography, computer science, and information engineering. His landmark books are: *Spatial Analysis and Planning under Imprecision* (Elsevier, 1988), *Intelligent Spatial Decision Support Systems* (Springer-Verlag, 1997), and *Knowledge Discovery in Spatial Data* (Springer-Verlag, 2010).

Organization

ICCSA 2011 was organized by the University of Cantabria (Spain), Kyushu Sangyo University (Japan), the University of Perugia (Italy), Monash University (Australia) and the University of Basilicata (Italy).

Honorary General Chairs

Antonio Laganà	University of Perugia, Italy
Norio Shiratori	Tohoku University, Japan
Kenneth C.J. Tan	Qontix, UK

General Chairs

Bernady O. Apduhan	Kyushu Sangyo University, Japan
Andrés Iglesias	University of Cantabria, Spain

Program Committee Chairs

Osvaldo Gervasi	University of Perugia, Italy
David Taniar	Monash University, Australia

Local Arrangements Chairs

Andrés Iglesias	University of Cantabria, Spain (Chair)
Akemi Gálvez	University of Cantabria, Spain
Jaime Puig-Pey	University of Cantabria, Spain
Angel Cobo	University of Cantabria, Spain
José L. Montaña	University of Cantabria, Spain
César Otero	University of Cantabria, Spain
Marta Zorrilla	University of Cantabria, Spain
Ernesto Anabitarte	University of Cantabria, Spain
Unal Ufuktepe	Izmir University of Economics, Turkey

Workshop and Session Organizing Chair

Beniamino Murgante	University of Basilicata, Italy

International Liaison Chairs

Jemal Abawajy	Deakin University, Australia
Marina L. Gavrilova	University of Calgary, Canada
Robert C.H. Hsu	Chung Hua University,Taiwan
Tai-Hoon Kim	Hannam University, Korea
Takashi Naka	Kyushu Sangyo University, Japan

Awards Chairs

Wenny Rahayu	LaTrobe University, Australia
Kai Cheng	Kyushu Sangyo University, Japan

Workshop Organizers

Approaches or Methods of Security Engineering (AMSE 2011)

Tai-hoon Kim	Hannam University, Korea

Approximation, Optimization and Applications (AOA 2011)

Ana Maria A.C. Rocha	University of Minho, Portugal
Maria Irene Falcao	University of Minho, Portugal

Advances in Web–Based Learning (AWBL 2011)

Mustafa Murat Inceoglu	Ege University (Turkey)

Computational Aspects and Methods in Renewable Energies (CAMRE 2011)

Maurizio Carlini	University of Tuscia, Italy
Sonia Castellucci	University of Tuscia, Italy
Andrea Tucci	University of Tuscia, Italy

Computer–Aided Modeling, Simulation, and Analysis (CAMSA 2011)

Jie Shen	University of Michigan, USA

Computer Algebra Systems and Applications (CASA 2011)

Andrés Iglesias	University of Cantabria (Spain)
Akemi Gálvez	University of Cantabria (Spain)

Computational Design for Technology–Enhanced Learning: Methods, Languages, Applications and Tools (CD4TEL 2011)

Michael Derntl	University of Vienna, Austria
Manuel Caeiro-Rodriguez	University of Vigo, Spain
Davinia Hernandez-Leo	Universitat Pompeu Fabra, Spain

Computational Geometry and Applications (CGA 2011)

Marina L. Gavrilova	University of Calgary, Canada

Computer Graphics and Virtual Reality (CGVR 2011)

Osvaldo Gervasi	University of Perugia, Italy
Andrés Iglesias	University of Cantabria, Spain

Chemistry and Materials Sciences and Technologies (CMST 2011)

Antonio Laganà	University of Perugia, Italy

Consulting Methodology and Decision Making for Security Systems (CMDMSS 2011)

Sangkyun Kim	Kangwon National University, Korea

Cities, Technologies and Planning (CTP 2011)

Giuseppe Borruso	University of Trieste, Italy
Beniamino Murgante	University of Basilicata, Italy

Cloud for High–Performance Computing (C4HPC 2011)

Andrés Gomez	CESGA, Santiago de Compostela, Spain
Osvaldo Gervasi	University of Perugia, Italy

Future Information System Technologies and Applications (FISTA 2011)

Bernady O. Apduhan	Kyushu Sangyo University, Japan
Jianhua Ma	Hosei University, Japan
Qun Jin	Waseda University, Japan

Geographical Analysis, Urban Modeling, Spatial Statistics (GEOG-AN-MOD 2011)

Stefania Bertazzon	University of Calgary, Canada
Giuseppe Borruso	University of Trieste, Italy
Beniamino Murgante	University of Basilicata, Italy

International Workshop on Biomathematics, Bioinformatics and Biostatistics (IBBB 2011)

Unal Ufuktepe	Izmir University of Economics, Turkey
Andrés Iglesias	University of Cantabria, Spain

International Workshop on Collective Evolutionary Systems (IWCES 2011)

Alfredo Milani	University of Perugia, Italy
Clement Leung	Hong Kong Baptist University, China

Mobile Communications (MC 2011)

Hyunseung Choo	Sungkyunkwan University, Korea

Mobile Sensor and Its Applications (MSA 2011)

Moonseong Kim	Korean Intellectual Property Office, Korea

Mobile Systems and Applications (MoSA 2011)

Younseung Ryu	Myongji University, Korea
Karlis Kaugars	Western Michigan University, USA

Logical, Scientific and Computational Aspects of Pulse Phenomena in Transitions (PULSES 2011)

Carlo Cattani	University of Salerno, Italy
Cristian Toma	Corner Soft Technologies, Romania
Ming Li	East China Normal University, China

Resource Management and Scheduling for Future–Generation Computing Systems (RMS 2011)

Jemal H. Abawajy	Deakin University, Australia

Remote Sensing Data Analysis, Modeling, Interpretation and Applications: From a Global View to a Local Analysis (RS 2011)

Rosa Lasaponara	IRMMA, CNR, Italy
Nicola Masini	IBAM, CNR, Italy

Symbolic Computing for Dynamic Geometry (SCDG 2011)

Francisco Botana	Vigo University, Spain

Software Engineering Processes and Applications (SEPA 2011)

Sanjay Misra	Atilim University, Turkey

Software Quality (SQ 2011)

Sanjay Misra	Atilim University, Turkey

Tools and Techniques in Software Development Processes (TTSDP 2011)

Sanjay Misra	Atilim University, Turkey

Virtual Reality in Medicine and Surgery (VRMS 2011)

Giovanni Aloisio	University of Salento, Italy
Lucio T. De Paolis	University of Salento, Italy

Wireless and Ad-Hoc Networking (WADNet 2011)

Jongchan Lee	Kunsan National University, Korea
Sangjoon Park	Kunsan National University, Korea

WEB 2.0 and Social Networks (Web2.0 2011)

Vidyasagar Potdar	Curtin University of Technology, Australia

Workshop on Internet Communication Security (WICS 2011)

Josè Maria Sierra Camara	University of Madrid, Spain

Wireless Multimedia Sensor Networks (WMSN 2011)

Vidyasagar Potdar	Curtin University of Technology, Australia
Yan Yang	Seikei University, Japan

Program Committee

Jemal Abawajy	Deakin University, Australia
Kenneth Adamson	Ulster University, UK
Michela Bertolotto	University College Dublin, Ireland
Sandro Bimonte	CEMAGREF, TSCF, France
Rod Blais	University of Calgary, Canada
Ivan Blecic	University of Sassari, Italy
Giuseppe Borruso	Università degli Studi di Trieste, Italy
Martin Buecker	Aachen University, Germany
Alfredo Buttari	CNRS-IRIT, France
Yves Caniou	Lyon University, France
Carlo Cattani	University of Salerno, Italy
Mete Celik	Erciyes University, Turkey

Alexander Chemeris	National Technical University of Ukraine "KPI", Ukraine
Min Young Chung	Sungkyunkwan University, Korea
Rosa Coluzzi	National Research Council, Italy
Stefano Cozzini	National Research Council, Italy
Josè A. Cardoso e Cunha	Universidade Nova de Lisboa, Portugal
Alfredo Cuzzocrea	University of Calabria, Italy
Frank Dévai	London South Bank University, UK
Rodolphe Devillers	Memorial University of Newfoundland, Canada
Pasquale Di Donato	Sapienza University of Rome, Italy
Carla Dal Sasso Freitas	UFRGS, Brazil
Prabu Dorairaj	NetApp, India/USA
Cherry Liu Fang	U.S. DOE Ames Laboratory, USA
Josè-Jesus Fernandez	National Centre for Biotechnology, CSIS, Spain
Francesco Gabellone	National Research Council, Italy
Akemi Galvez	University of Cantabria, Spain
Marina Gavrilova	University of Calgary, Canada
Jerome Gensel	LSR-IMAG, France
Andrzej M. Goscinski	Deakin University, Australia
Shanmugasundaram Hariharan	B.S. Abdur Rahman University, India
Hisamoto Hiyoshi	Gunma University, Japan
Fermin Huarte	University of Barcelona, Spain
Andres Iglesias	University of Cantabria, Spain
Peter Jimack	University of Leeds, UK
Qun Jin	Waseda University, Japan
Farid Karimipour	Vienna University of Technology, Austria
Baris Kazar	Oracle Corp., USA
Ivana Kolingerova	University of West Bohemia, Czech Republic
Dieter Kranzlmueller	LMU & LRZ Munich, Germany
Domenico Labbate	University of Basilicata, Italy
Antonio Laganà	University of Perugia, Italy
Rosa Lasaponara	National Research Council, Italy
Maurizio Lazzari	National Research Council, Italy
Cheng Siong Lee	Monash University, Australia
Sangyoun Lee	Yonsei University, Korea
Jongchan Lee	Kunsan National University, Korea
Clement Leung	Hong Kong Baptist University, Hong Kong
Chendong Li	University of Connecticut, USA
Ming Li	East China Normal University, China
Xin Liu	University of Calgary, Canada
Savino Longo	University of Bari, Italy
Tinghuai Ma	NanJing University of Information Science and Technology, China
Sergio Maffioletti	University of Zurich, Switzerland

Sponsoring Organizations

ICCSA 2011 would not have been possible without tremendous support of many organizations and institutions, for which all organizers and participants of ICCSA 2011 express their sincere gratitude:

- The Department of Applied Mathematics and Computational Sciences, University of Cantabria, Spain
- The Department of Mathematics, Statistics and Computation, University of Cantabria, Spain
- The Faculty of Sciences, University of Cantabria, Spain
- The Vicerrector of Research and Knowledge Transfer, University of Cantabria, Spain
- The University of Cantabria, Spain
- The University of Perugia, Italy
- Kyushu Sangyo University, Japan
- Monash University, Australia
- The University of Basilicata, Italy
- Cantabria Campus Internacional, Spain
- The Municipality of Santander, Spain
- The Regional Government of Cantabria, Spain
- The Spanish Ministry of Science and Innovation, Spain
- GeoConnexion (http://www.geoconnexion.com/)
- Vector1 Media (http://www.vector1media.com/)

Table of Contents – Part V

Workshop on Mobile Communications (MC 2011)

Workshop on Software Quality (SQ 2011)

Workshop on Software Engineering Processes and Applications (SEPA 2011)

Session on International Workshop on Biomathematics, Bioinformatics and Biostatistics (IBBB 2011)

General Tracks

Selective Placement of High-Reliability Operation in Grid-Style Wireless Control Networks*

Junghoon Lee[1], Gyung-Leen Park[1,**], and Ho-Young Kwak[2]

[1] Dept. of Computer Science and Statistics, [2] Dept. of Computer Engineering
Jeju National University
690-756, Jeju Do, Republic of Korea
{jhlee,glpark,kwak}@jejunu.ac.kr

Abstract. This paper handles the problem of how to select the node which runs the reliability operation in the WirelessHART network having a grid topology. The split-merge operation can be placed on any rectangular path to improve the transmission success ratio, dynamically adapting the route according to the current link condition, and thus overcoming the inflexibility of the fixed schedule. Based on the observation that almost every link is used in the sensor-to-controller route and the corresponding slot schedule, the proposed scheme gives precedence to the nodes having a high error level. The node error level can be calculated by averaging the error rate of all links emanating from the node. The performance evaluation result measured by extensive simulation using a discrete event scheduler demonstrates that with 75 % split-merge operations, we can achieve over 90 % success ratio when the grid dimension is 5, compared with full allocation.

Keywords: wireless process control, WirelessHART, split-merge operation, selective allocation, reliability improvement.

1 Introduction

Basically, wireless communication can fully take advantage of elimination of installation cost, enhancement of flexibility, ease of maintenance, and the like. While the wireless network keeps replacing the wired network in many areas, it has been questionable whether the wireless network can be deployed in industrial process control due to reliability, security, speed, and battery life [1]. Hence, emerging technologies are challenging those problems, pursuing compatibility with the existing wired process control equipment. In the mean time, ISA (International Society for Automation) approved the ISA-100.11a wireless standard called *Wireless Systems for Industrial Automation: Process Control and*

* This research was supported by the MKE (The Ministry of Knowledge Economy), through the project of Region technical renovation, Republic of Korea.

** Corresponding author.

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 1–10, 2011.

Related Applications, to the end of meeting needs of industrial wireless users and operators of low-energy fixed, portable, and moving devices [2]. It defines specifications for the protocol suite, system management, gateways, and security for low-data-rate wireless connectivity.

Based on the proven and widely accepted (wireline) HART technology, the recently established WirelessHART standard provides a robust wireless protocol for distributed process control applications [3]. Its key advantages lie in its design for reliability, security, and efficient power management. In particular, WirelessHART is designed to tolerate the transient instability of wireless channels by adopting advanced techniques in mesh networking, channel hopping, and time-synchronized messaging. It specifies a new data link and network layers, but compatible with the wired HART protocol from the transport layer. This protocol has been implemented and is being deployed in the process control market [4], while the primary application area of this technology includes condition and performance monitoring of critical equipment.

All WirelessHART devices have routing functions, making the WirelessHART network simple and self-organizing [5]. Particularly, it employs graph routing and source routing, but in the process control, graph routing is preferred. This routing scheme transmits the message along the path predefined by a network manger according to the specific routing algorithm. To cope with the channel error, path redundancy is indispensable. Hence, a graph route includes one or more different paths. However, the problems on how many redundant paths are available and which path to select as the secondary path depend on the underlying network topology and the target industrial application. In our previous work, we have proposed a routing and slot allocation scheme for a grid-topology network running WirelessHART. It considered every rectangular path as a pair of primary and secondary paths. Based on the so-called *split-merge* (SM) operation, a node selects a path dynamically according to the current channel condition.

The SM operation can enhance the message delivery ratio without delaying the end-to-end transmission time or installing a non-compliant circuit [6]. However, it makes the sender and the receiver more wake up to wait for the possible redundant transmission, consuming the battery power. The overhead depends on the number of rectangular paths. For a $N \times N$ grid, $(N-1) \times (N-1)$ rectangles exist. This paper asserts that not all rectangles need to perform the dynamic path selection which is part of the SM operation. In the monitoring application, every node sends sensor messages to a controller. In such many-to-one paths, two or more paths overlap indispensably. If the overlapped part suffers from the poor transmission channel quality, it is necessary to place an additional compensation scheme such as the SM operation or more powerful transmitter. In this regard, this paper is to propose a partial allocation scheme which finds the weak point of the given wireless process control network having a grid-topology. By this, the network is expected to save power consumption with the reliability level comparable to the full SM allocation.

This paper is organized as follows: After issuing the problem in Section 1, Section 2 describes the background of this paper and related work. Section 3 proposes the partial allocation of split-merge operations on the grid-topology wireless network. After the performance measurement results are discussed in Section 4, Section 5 summarizes and concludes this paper with a brief introduction of future work.

2 Background and Related Work

2.1 WirelessHART

The WirelessHART standard supports multiple messaging modes including one-way publishing of process and control values, spontaneous notification by exception, ad-hoc request/response, and auto-segmented block transfers of large data sets [7]. According to the standard, the WirelessHART network consists of field devices, gateways, and a network manager. Wireless field devices are connected to process or plant equipment with a WirelessHART adapter attached. Gateways enable communication between these devices and host applications connected to a high-speed backbone or other existing plant communications network. A network manager is responsible for configuring the network, scheduling communications between devices, managing message routes, and monitoring network health. The network manager can be integrated into the gateway, host application, or process automation controller. Just like Zigbee and Bluetooth, WirelessHART operates on the IEEE 802.15.4 GHz radioband PHY and MAC [7]. It includes 16 frequency channels spaced by 5 MHz guard band.

For predictable network access, the link layer is based on TDMA style access mechanism on top of the time synchronization mechanism carried out continuously by means of MAC PDUs [4]. TDMA eliminates collisions and reduces the power consumption as the device has to keep the radio on only during the time slots it is associated with, as contrast to the Zigbee case where CSMA/CA makes each receiver keep alive for a large part of the frame [8]. The time axis is divided into 10 ms time slots and a group of consecutive slots is defined to be a superframe, which repeats during the network operation time. To meet the robustness requirement of industrial applications, the network manager centrally decides routing and communication schedules. This schedule can be propagated to the field devices in priori of system operation or updated via the channel reserved for this purpose. For each path, each (sender, receiver) pair is assigned to a time slot.

WirelessHART has many reliability enhancement features. First of all, frequency hopping and retransmissions alleviate the effects of temporal and frequency interference. Here, a retransmission will occur on a different frequency. Second, each time slot includes CCA (Clear Channel Assessment), which takes just a few bit time. Due to the half-duplex nature of current wireless systems, collision detection can't be available. Hence, automatic CCA before each transmission and channel blacklisting can also avoid specific area of interference and also to minimize interference to others. The 802.15.4 standard specifies that CCA may be performed using energy detection, preamble detection, or a combination of the two. In addition, there are reliable channel estimation methods available for the MAC layer to obviate the erroneous communication over the unclear channel. Channel probing can be accomplished by CCA, RTS/CTS, and so on [9]. Here, we assume that there is no false alarm, as the correctness of channel probing is not our concern. Besides, WirelessHART has many features useful for process control such as node join and leave operations in addition to time synchronization. For detailed description, refer to Song's work [4].

2.2 Channel Schedule

Communication on the grid topology is extensively studied in the wireless mesh network area, and it mainly focuses on efficient routing inside the grid to achieve various optimization goals, like minimizing end-to-end delays, ensuring fairness among nodes, and minimizing power consumption [10]. The channel assignment problem is NP-hard [11], so even for the case of no interference in the network, the optimal schedule cannot be computed in polynomial time. Many greedy algorithms are applied for the static channel assignment including the well-known coloring algorithm. That is, colors or time slots must be assigned such that no two adjacent nodes in the network have the same color. For slot-based access, most schemes focus on how to assign a color on the diverse topology and how to cope with the topology change. Anyway, existing schemes have not dealt with the dynamic channel adaptation within a single slot and the subsidiary slot allocation.

N. Chang considers transmission scheduling based on optimal channel probing in a multichannel system, where each channel state is associated with a probability of transmission success and the sender just has partial information [12]. To complement the overhead and resource consumption of the channel probing procedure, this work proposes a strategy to decide the channel to probe based on a statistical channel model. Using this estimation, the sender or scheduler can pick the channel to use for transmission. Even though the concept of channel probing is very prospective, this scheme just pursues the dynamic operation of communication procedure, so it can't be applied to process control. Moreover, as it doesn't consider the route length but just the probability estimation on successful transmissions in the selection of an available channel, it lacks predictability.

A slot management on WirelessHART is considered along with a mathematical framework in terms of modeling and analysis of multi-hop communication networks [13]. This model allows us to analyze the effect of scheduling, routing, control decision, and network topology. In this model, each node has at least two neighbor choices to route a packet for any destination nodes. Hence, the time slot schedule must explicitly have two paths for each source and destination pair. That is, regardless of whether the primary path successfully transmits a message, the secondary route redundantly delivers the same message. This scheme can integrate any style of an alternative route such as a node-disjoint or link-disjoint path, but bandwidth waste is unavoidable and slot allocation is highly likely to get too complex.

3 Channel Scheduling Scheme

3.1 Grid Topology and Split-Merge Operations

Most modern cities have a Manhattan-style road network, and if a wireless node is placed at each crossing of the road network, for example, on the traffic light, the traffic light network has the grid topology as shown in Figure 1(a). In addition, many monitor and control systems including agricultural irrigation system and environmental data collecting system have sensors installed in the grid topology. A node, having at most 4 neighbors, and the directional antenna is desirable to narrow the interference area [14]. In this network, one node plays a role of central controller, and we assume that it is located at the left top corner of rectangular area, for this architecture makes the determination of communication schedule simple and systematic. In Figure 1(a), $N_{0,0}$ is the controller node. It can be generalized by the grid partition as shown in Figure 1(b), where 4 quadrants can be found irrespective of controller location. Each of quadrants can be mapped or transformed to the network shown in Figure 1(a) by making the controller be placed at the left top corner.

As an example of a rectangular path, consider the downlink transmission from $N_{0,0}$ to $N_{1,1}$ which has two 2-hop paths, namely, $N_{0,0} \rightarrow N_{1,0} \rightarrow N_{1,1}$, and $N_{0,0} \rightarrow N_{0,1} \rightarrow N_{1,1}$. As they are symmetric, we just consider the downlink case. The schedule will include ($N_{0,0} \rightarrow N_{1,0}$) and ($N_{1,0} \rightarrow N_{1,1}$) at time slots, say, t and $t+1$, respectively. The sender, $N_{0,0}$, senses a channel status for $V_{1,0}$ at slot t. If it is clear, it sends to $N_{1,0}$ according to the original schedule. Otherwise, instead of discarding the transmission, it sends to $N_{0,1}$ after switching the channel (*Split* operation). In the receiver side, in slot t, $N_{0,1}$ (the receiver on the alternative path) as well as $N_{1,0}$ (the receiver on the primary path) must listen to the channel simultaneously, and this is the overhead brought for enhanced reliability. At slot $t+1$, either $N_{1,0}$ or $N_{0,1}$ can possibly send a packet to $N_{1,1}$. $N_{1,1}$ first tries to receive via $N_{1,0}$, as it is the primary sender. If the packet arrives, it receives as scheduled. Otherwise, namely, the message does not arrive until the $TsRxOffset+TsError$, the receiver immediately switches to $H_{1,1}$ and receives the packet ($Merge$ operation). This operation is summarized in Figure 2 and refer to [6] for more detail.

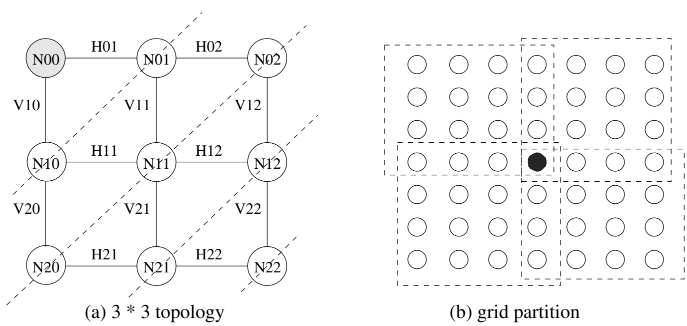


Fig. 1. Traffic light network

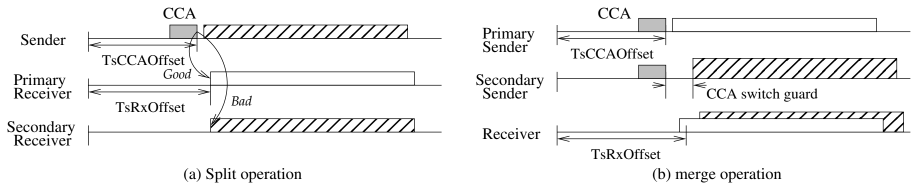


Fig. 2. Split-merge operation

3.2 Partial Allocation

For the given link error rate distribution, we can decide the route from each sensor node to the controller using the well-known Dijskstra algorithm. For the diagonal path capable of placing a SM operation, say, $N_{i,j} \rightarrow N_{i+1,j+1}$, the virtual link is added to take into account the improved success probability. This section focuses on where to put SM operations for the given number of them, not placing all of possible rectangles. They can be placed according to the node location in the grid, for example, near the controller, far away from the controller, along the fringe, and so on. If the link closer to the controller has good quality, the probability of wasteful transmission decreases. On the long path, if the last hop fails, the successful transmissions in the earlier stage of the end-to-end path will be meaningless and the transmission power was wasted. If the error rate of a node farther away from the controller gets better, the node has the enhanced chance for a successful end-to-end transmission.

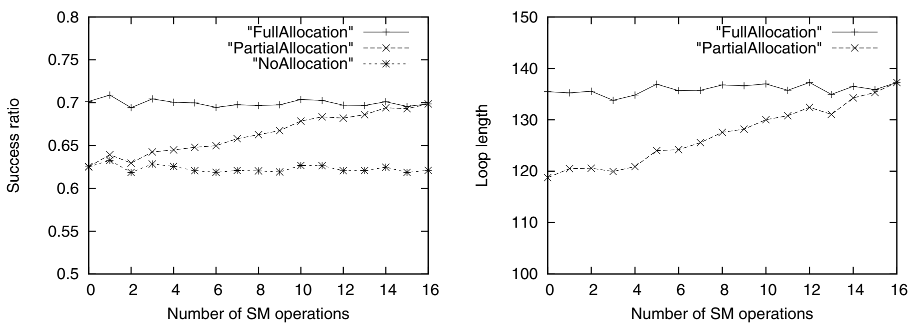
Among various options, based on the observation that almost every link is chosen at least once in the rout decision and slot allocation procedure, this paper is to put SM operations on the poor-quality link, mainly aiming at improving the end-to-end delivery ratio. The selection process begins with the calculation of node error level for each node. The node error level is decided by averaging the error rates of all links emanating from the node. A node in the grid can have 2, 3, or 4 links. From the node having the largest error level, the selection procedure checks if an SM operation can be placed. If the node is located at two fringes from the right bottom node, the SM operation cannot run on it, as there is no rectangular path. In this way, we can manage the number of SM nodes, making it possible to provide compatibility between SM nodes and general nodes as well as to select the node that need to be replaced by the high-cost and better error-resistant one.

4 Performance Measurement

This section measures and assesses the performance of the proposed selective placement scheme via simulation using SMPL, which provides a discrete event scheduler [15]. The simulation assumes that the link error rate distributes exponentially with the given average value, obeying the Guilbert-Elliot error model. The experiment mainly measures the success ratio and the loop length according to the link error rate and the grid dimension. The success ratio indicates the ratio of end-to-end successful transmissions. Generally, the closer to the controller, the higher the success ratio. For each parameter setting, we generate 500 link sets, measure the performance results, and average them. The graphs in this section have 2 or 3 curves. First, the curve marked with *Full Allocation* corresponds to the case every $(N-1) \times (N-1)$ rectangles runs SM operations in the $N \times N$ grid as in [6]. Second, *No Allocation* curve is the case no SM operation is placed in the grid. Third, *Partial Allocation* curve plots the performance of the selective allocation scheme proposed in Subsection 3.2.

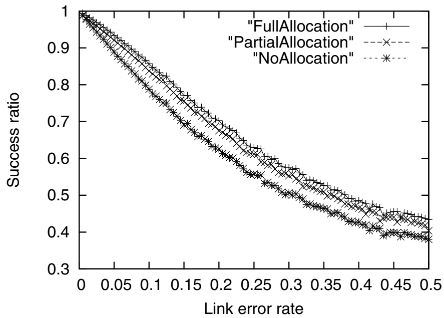
The first experiment measures the effect of the number of SM operations to both the success ratio and the loop length. The grid dimension is set to 5, while the average link error rate is set to 0.2. In the $N \times N$ grid, the number of SM operations can range from 0 to 16. As shown in Figure 3(a), the success ratio curves of full allocation and no allocation are almost flat, as they don't depend on the number of SM operations. The partial allocation curve meets no allocation curve when the number of SM operations is 0 while it meets the full allocation when the number is 16. For this parameter setting, the performance gap between full allocation and no allocation roughly stays around 8 %. The partial allocation shows almost same success ratio when the number of SM operations is 12, indicating that it is not necessary to make all nodes run the SM operation. Figure 3(b) plots the loop length of two cases, namely, full allocation and partial allocation. The larger loop length means that some end-to-end paths take detours having more hops, but lower transmission failure probability. It shows the similar pattern with Figure 3(a).

Figure 4 measures the effect of the average link error rate, changing it from 0 to 0.5. In this experiment, we set the grid dimension to 5 and the number of SM operations to 12. We can find out from Figure 4(a) that the selective allocation scheme shows the success ratio comparable to the full allocation regardless of the link error rate distribution. With 75 % SM operations, we can achieve over 90 % success ratio, compared with full allocation. The same success ratio, namely, 100% comparable to the full allocation, cannot be obtained as at least each link is generally included at least once in the route schedule. Figure 4(b) also plots

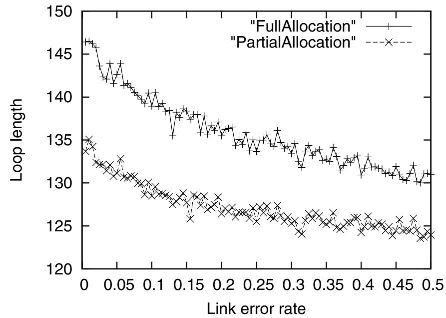


(a) success ratio measurement (b) loop length measurement

Fig. 3. Effect of the number of SM operations



(a) success ratio measurement



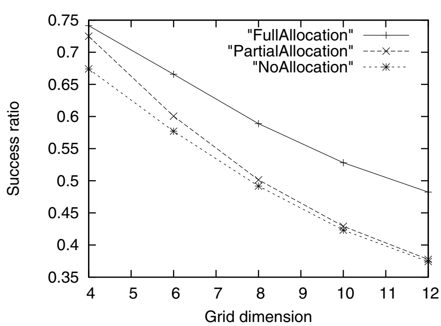
(b) loop length measurement

Fig. 4. Effect of link error rate

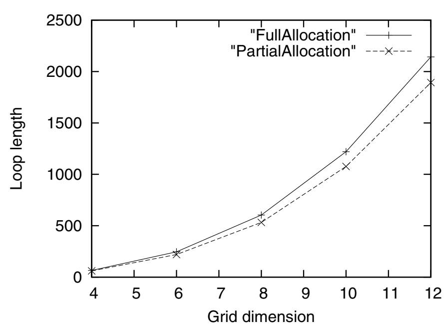
the loop length according to the link error rate for full and partial allocation schemes. The selective scheme achieves 4.5 % loop length reduction, sacrificing the success ratio just by 1.1 %.

Figure 5 plots the effect of the grid dimension to the success ratio and the loop length. This experiment changes the dimension from 4 to 12 with the link error rate fixed to 0.2 and the number of SM operations to 75 % of all $(N-1) \times (N-1)$ candidate nodes. Figure 5(a) shows that the proposed scheme works well on the small grid, but it goes closer to the no allocation case in the large grid. We think that some paths are vulnerable to the link error and their success ratio cannot be complemented without full allocation. After all, the success ratio more depends on the absolute number of SM operations. In addition, the ratio of loop lengths for two cases remains constant as shown in Figure 5(b).

Finally, Figure 6 measures the per-node statistics. To begin with, nodes are numbered from 1 to 15 in the row-major order to represent in the graph generated by *gnuplot*. For example, in the 5 × 5 grid, $N_{0,3}$ corresponds to 3 and $N_{1,1}$ to 6, respectively, in the x-axis of graphs in Figure 6. Here, the grid dimension

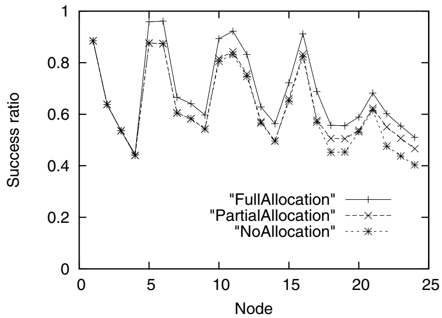


(a) success ratio measurement

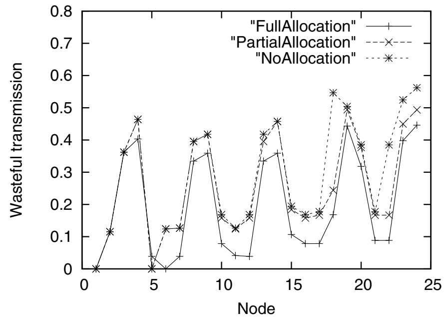


(b) loop length measurement

Fig. 5. Effect of grid dimension



(a) success ratio measurement



(b) wasteful transmission measurement

Fig. 6. Per-node statistics

is 5 and the average link error rate is 0.2. For the nodes on the diagonal link connecting node 0 and node 24 generally shows better success ratio. For the nodes on the fringe line connected to the controller, the success ratio is almost same. However, when the direct link has low success ratio estimation, they can also take advantage of SM operations, selecting the route having more hops but a lower message failure probability. In addition, Figure 6(b) plots the wasteful transmission, which takes place when the succeeding node fails to deliver the message even if a node successfully relays a message. In this case, the power consumed in the slot is meaningless.

5 Conclusions

This paper has addressed the problem of how to select the node which runs the SM operation in the WirelessHART network having a grid topology. It is aiming at demonstrating that partial reliability operation allocation can achieve the performance comparable to the full allocation. The node empowered with the SM operation can enhance the delivery success ratio by dynamically adapting the route according to the current link condition, and thus overcoming the fixed schedule generally employed in the slot-based MAC. However, it brings additional power consumption caused by redundant listening and transmission. Based on the observation that almost every link is used in the sensor-to-controller route and the corresponding slot schedule, the proposed scheme selects the nodes according to the node error level which is obtained by averaging the error rate of all links emanating from the node. The performance evaluation result measured by simulation using a discrete event scheduler demonstrates that with 75 % SM operations, we can achieve over 90 % success ratio, compared with full allocation.

As future work, we will design a routing scheme and the corresponding channel assignment method for the secondary schedule to take into the difference in the error characteristics of each channel. This work includes an issue such as where to put the split-merge operation when more than one alternative path is available.

References

1. Chen, D., Nixon, M., Aneweer, T., Shepard, R., Mok, A.: Middleware for wireless process control systems. In: WACERT (2004)
2. Raza, S., Voigt, T.: Interconnecting WirelessHART and legacy HART networks. In: International Conference on Distributed Computing in Sensor Systems Workshops, pp. 1–8 (2010)
3. Hart Communication Foundation, Why WirelessHART^{TM}? The Right Standard at the Right Time (2007), http://www.hartcomm2.org
4. Song, S., Han, S., Mok, A., Chen, D., Nixon, M., Lucas, M., Pratt, W.: WirelessHART: Applying Wireless Technology in Real-Time Industrial Process Control. In: The 14th IEEE Real-Time and Embedded Technology and Applications Symposium, pp. 377–386 (2008)
5. Han, S., Song, J., Zhu, X., Mok, A.K., Chen, D., Nixon, M.: Pratt. W., Gondhalekar, V.: Wi-HTest: Compliance test suite for diagnosing devices in real-time WirelessHART network. In: The 15th IEEE Real-Time and Embedded Technology and Applications Symposium, pp. 327–336 (2009)
6. Lee, J., Song, H., Mok, A.K.: Design of a reliable communication system for grid-style traffic control networks. In: The 16th IEEE Real-Time and Embedded Technology and Applications Symposium, pp. 133–142 (2010)
7. IEC/PAS 62591: Industrial communication networks - Fieldbus specifications - WirelessHART communication network and communication profile (2008)
8. Lennvall, T., Svensson, S., Hekland, F.: A comparison of WirelessHART and Zigbee for industrial applications. In: IEEE International Workshop on Factory Communication Systems, pp. 85–88 (2008)
9. Ramchandran, I., Roy, S.: Clear channel assessment in energy-constrained wideband wireless networks. IEEE Wireless Magazine, 70–78 (2007)
10. Kodialam, M., Nandagopal, T.: Characterizing the capacity region in multi-radio multi-channel wireless mesh networks. In: ACM MobiCom, pp. 73–87 (2005)
11. Wan, P., Huang, S., Wang, L., Wan, Z., Jia, X.: Minimum latency aggregation scheduling in multihop wireless networks. In: MobiHoc, pp. 185–193 (2009)
12. Chang, N., Liu, M.: Optimal channel probing and transmission scheduling for opportunistic spectrum access. In: ACM International Conference on Mobile Computing and Networking, pp. 27–38 (2007)
13. Alur, R., D'Innocenzo, A., Pappas, G., Weiss, G.: Modeling and analysis of multihop control networks. In: The 15th IEEE Real-Time and Embedded Technology and Applications Symposium, pp. 223–232 (2009)
14. Dai, H., Ng, K., Wu, M.: An overview of MAC protocols with directional antennas in wireless ad hoc networks. In: International Conference on Computing in the Global Information Technology, pp. 84–91 (2006)
15. MacDougall, M.: Simulating Computer Systems: Techniques and Tools. MIT Press, Cambridge (1987)

Adaptive Clustering Method for Femtocells Based on Soft Frequency Reuse*

Young Min Kwon, Bum-Gon Choi, Sueng Jae Bae, and Min Young Chung**

School of Information and Communication Engineering Sungkyunkwan University
300, Chunchun-dong, Jangan-gu, Suwon, Gyeonggi-do, 440-746, Republic of Korea
{ko116,gonace,noooi,mychung}@ece.skku.ac.kr

Abstract. Femtocell base station (FBS) is one solution to improve performance of indoor and outdoor user equipments (UEs) with low cost and low complexity. To gain the benefit of deployment of FBS, we consider a frequency assignment method to mitigate cross-tier interference between macrocell and femtocell. Previous studies proposed the channel assignment schemes where macrocell base stations (MBSs) and FBSs orthogonally use subbands. However, they have low efficiency in terms of frequency reuse and high complexity due to exchanging information of FBSs via network. Therefore, we consider the FBS with soft frequency reuse (SFR) scheme to improve the efficiency of frequency reuse. For performance improvement of macrocell UEs (MUEs), we propose a method to cluster FBSs with same SFR pattern. Since the proposed clustering scheme utilizes self-organization network (SON) technique without any information exchange of FBSs, it can improve the performance of MUEs with low complexity. By simulations, we confirm that FBSs with proposed clustering method mitigate cross-tier interference in downlink and increase the throughputs of MUEs.

Keywords: Femtocell, SFR, FFR, SON, cross-tier interference, cluster.

1 Introduction

3GPP long term evolution advanced (LTE-A) system considers the deployment of femtocell base station (FBS) in order to satisfy requirements of the IMT-advanced [1]. FBS is one of the attractive solutions to improve performance of indoor user equipment (UE) within coverage area of macrocell base station (MBS) with low cost and low complexity. Since a FBS uses same air interface with an MBS, it can simply take over the indoor UEs' traffic served by MBSs through handover procedure defined in the standard. Therefore, FBSs can provide high data rate service for indoor UEs connected to FBSs and improve system capacity [2].

* This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency) (NIPA-2011-(C1090-1111-0005)).

** Corresponding author.

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 11–21, 2011.

As we have mentioned above, the deployment of FBSs can improve performance of not only UEs connected to MBS but also indoor UEs connected to FBS. However, if so many FBSs are deployed in the coverage area of an MBS and use the same frequency band with MBSs, the benefit of FBS would be reduced due to the cross-tier interference between macrocell and femtocell. The cross-tier interference means that different kinds of cells affect one another. In the cross-tier interference, victim and aggressor are alternated according to the direction of link as shown in Figure 1. Especially, if a macro UE (MUE) is located at the edge of MBS and close to FBSs, the performance of MUE and FBS is severely decreased by the cross-tier interference. In downlink, the MUE (victim) suffers the undesirable signaling from nearby FBS (aggressor) and vice versa in uplink. When FBSs are deployed, therefore, spectrum allocation should be considered in order to solve the degradation of performance caused by the cross-tier interference.

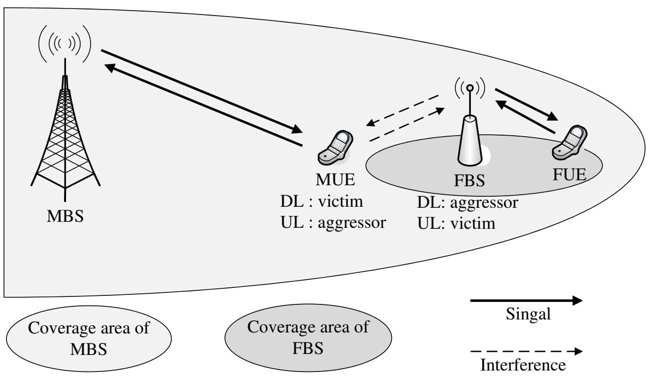


Fig. 1. Cross-tier interference between macrocell and femtocell

There are two approaches in spectrum assignment, orthogonal channel assignment and co-channel assignment [3]. Orthogonal channel assignment eliminates the cross-tier interference by dividing the system frequency band into two subbands and by orthogonally allocating the different subbands to macrocell and femtocells. In the co-channel assignment, on the other hand, macrocell and femtocell can use whole frequency band. In terms of utilization, the orthogonal channel assignment would use frequency band inefficiently because the system bandwidth is strictly divided. Thus, we expect that most of operators would be considered the use of co-channel assignment in FBS to increase the reuse of expensive licensed frequency band and reduce communication cost [4].

In this paper, we consider the frequency reuse scheme based on the co-channel assignment to reduce cross-tier interference. Previous frequency reuse schemes for femtocell mostly consider the orthogonal channel assignment [5][6][7]. In these schemes, cross-tier interference between macrocell and femtocell located in macrocell is eliminated by using frequency band which is not used in the macrocell. However, these schemes based on orthogonal channel assignment would be hard to solve the problem of low frequency utilization. Therefore, we adopt soft frequency reuse (SFR) scheme as the cell planning method of FBS and propose the clustering method where FBSs with same SFR pattern group together in order to not only reduce cross-tier interference but also increase frequency utilization. An outline of the rest of this paper is as follows.

In Section 2, we introduce a previous frequency reuse scheme for femtocell and self-organization network technique. Then, we explain the effect of FBSs with SFR on MUEs and propose the selection method of SFR pattern in Section 3. The result and analysis of simulation are presented in Section 4. Finally, conclusions are drawn in Section 5.

2 Preliminaries

Deployment of the FBS can improve the performance of indoor UE connected to FBS as well as remaining outdoor UE connected to MBS in downlink. However, the benefit can be maximized when cross-tier interference is sophisticatedly coordinated. In this chapter, we introduce spectrum allocation scheme and self-organization network (SON) technique to mitigate cross-tier interference.

There are two spectral allocation cases between macrocell and femtocell, orthogonal channel assignment and co-channel assignment [3]. In the orthogonal channel assignment, total system frequency band is divided into two subbands and allocated to macrocell and femtocell. Since macrocell and femtocell orthogonally use different subband, the cross-tier interference can be completely eliminated. However, since macrocell and femtocell can not use whole system frequency band, the orthogonal channel assignment is inefficient in term of frequency reuse and has relatively low cell throughput to co-channel assignment. In co-channel assignment, on the other hand, macrocell and femtocell can unrestrictedly use whole system frequency band. Therefore, it provide higher frequency reuse and cell throughput as compared to the orthogonal channel assignment.

Self-organization network (SON), which is one of the femtocell technical issues, helps FBS to optimally tunes its parameters by measuring the channel condition by itself. The base stations of conventional cellular system are deliberately installed and configured by operator while FBSs are deployed in arbitrary place by end-users. Therefore, SON function are demanded to FBS in order to autonomously configure and tune its parameters. For SON, a FBS can collect measurement information through active femtocell UE (FUE) connected to FBS or a downlink (DL) receiver function within the FBS itself [8]. In the measurement via FUE, a FBS obtains channel quality identifier (CQI) measured in the location of it's FUE and sets its parameter according to the measurement results in CQI. on the other hand, a FBS itself can measure channel condition in the measurement via DL receiver function. In terms of accuracy, the measurement via DL receiver function is good since there may be significant difference between the radio frequency condition of FUE and FBS in the measurement through FUE [8].

Previous frequency reuse schemes are divided into static and dynamic scheme. The static scheme fixes the range of frequency bands assigned to macrocell and femtocell [5]. In [5], the authors consider that MBSs use Fractional Frequency Reuse (FFR) and FBSs utilize frequency bands not used in macrocell. An MBS with FFR divides whole frequency band into several subbands and allocates different subbands to MUEs according to their locations within the macrocell. The MBS allocates communal subband used in all MBSs to MUEs located in center of macrocell. While, it assigns subbands orthogonally used between adjacent cell to MUEs in edge of macrocell to reduce co-tier interference between macrocell. To avoid interference with MUEs, FBSs within the macrocell assign subbands not used due to edge MUEs of adjacent macrocell to FUEs. In downlink, the static FFR scheme eliminates cross tier interference between MUEs served by a MBS and FBS in coverage area of the MBS. However, since the static FFR scheme is inefficient in term of frequency reuse, it would provide low capacity in macrocell and femtocell.

To improve the frequency reuse, the dynamic schemes are proposed [6][7]. In the dynamic schemes, FBSs selects subbands to optimize the throughput of each cell according to its channel condition. In [6], the range of frequency bands for MBS and FBS are determined according to expected throughput per UE and quality of service (QoS) requirement of each cell. In the proposed scheme, the throughput of each cell is estimated by the MUE and FUE measuring the interference from neighboring cells. Then, whole frequency band is divided into two subbands for MBS and FBS under constraint, in which MBS and FBS should guarantee a minimum throughput to UEs. To maximize frequency reuse, authors adopt the frequency ALOHA scheme where FBSs randomly select subbands within the range of frequency band for femtocell.

Besides, dynamic scheme to find optimal subband between BSs by using jamming graph is proposed in [7]. In the proposed scheme, after each BS negotiates the use of a certain subband based on its quantified benefit estimated by jamming graph, it uses optimal subbands. The jamming graph consists of a node and a edge denoting an active BSs and jamming condition between two BSs, respectively. In the jamming graph, jamming condition means the channel gain difference between signal and interference. The mentioned dynamic schemes increase efficiency of frequency reuse by adaptively assigning the subband according to channel condition. However, since optimal solution of these dynamic schemes can be given by exchanging information between each BS via wired or wireless backhaul, the performance would depend on the number of messages and the latency in backhaul [8].

3 Proposed Scheme

The static and dynamic frequency partitioning schemes have drawbacks for low efficiency of frequency reuse and network sensitivity, respectively. Therefore, we consider the use of SFR in cellular system where MBSs utilize whole system band with same transmission power in order to increase the frequency reuse efficiency.

In addition, we consider that FBSs itself select a SFR pattern by using SON technique without any information exchange between FBSs so that they with same SFR pattern cluster together.

3.1 Effect of Femtocell Base Station with Soft Frequency Reuse

We consider a general cellular system where MBSs utilize entire system band with same transmission power. In proposed scheme, we assume that all FBSs adopt the SFR scheme and the serving MBS allocates the radio resources by using the proportional fair (PF) scheduling. A FBS with SFR divides whole frequency band into three subbands and tunes transmission power in each subband according to the SFR pattern selected by the FBS itself. In a SFR patten, FBS transmits signal with high transmission power (P_{high}) in one subband and with low transmission power (P_{low}) in the remaining two subbands. The ratio between P_{high} and P_{low} is pre-determined as $\rho(=\frac{P_{high}}{P_{high}})$ and the sum of P_{high} and P_{low} is equal to the total transmission power of a FBS. As shown in Figure 2, when a MUE locates close to FBS with a certain SFR pattern, the MUE suffers relatively low interference in range of subband where the FBS transmits signal with P_{low}. Then, the MUE reports the CQI including the high SINR value in range of subband to its serving MBS as Figure 3.

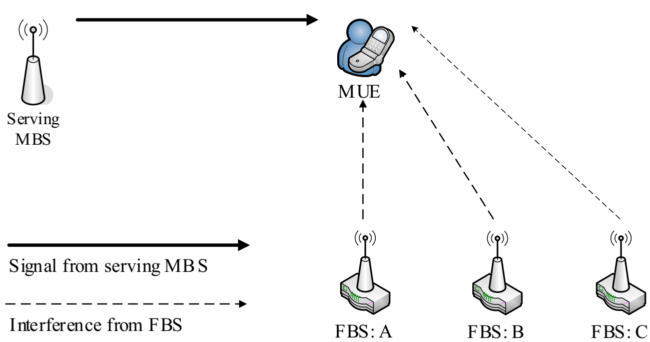


Fig. 2. Interference scenario of SFR femtocell

Up on receiving the CQI report with SINR value, the MBS estimates the achievable rate R of the MUE in sub-channel by mapping SINR of the MUE to MCS table. Then, the MBS allocates a radio resource to the MUE according to the given achievable rate R and average rate A of the MUE based on PF scheduling manner. If an MUE i has the largest $\frac{R_i}{A_i}$, a serving MBS assigns the sub-channel to the MUE i and calculates the average rate of all MUEs by using moving average with weight α.

$$A(t+1) = \begin{cases} (1-\alpha)\cdot A(t) + \alpha \cdot R, & \text{if scheduled,} \\ (1-\alpha)\cdot A(t), & \text{if not scheduled.} \end{cases}$$

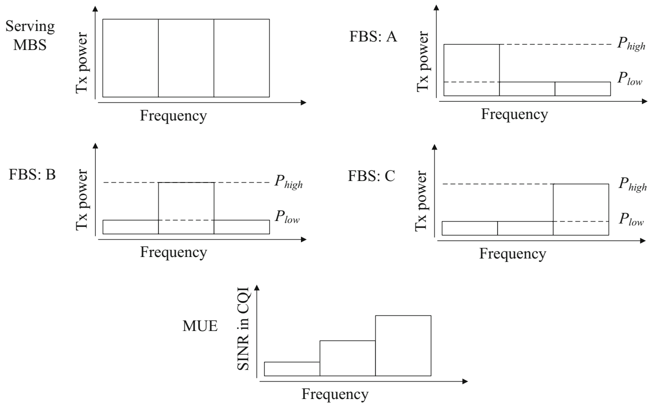


Fig. 3. SINR of MUE surrounding FBS with SFR

In PF scheduling, serving MBS mostly allocates subbands to MUEs with high SINR in range of the subband since R of MUE is determined by their SINR. Therefore, FBSs with SFR would increase the throughput of neighboring MUEs without additional interference coordination procedure between FBS and MUE.

3.2 SFR Pattern Selection Method

As we mentioned in Section 3.1, when a MBS uses the PF scheduling scheme, performance of MUE is improved due to the subband where FBSs with SFR use low transmission power P_{low}. When a cluster of FBSs use same SFR pattern, SINR of a MUE surrounded by the FBSs can be significantly increased in subbands where FBSs surrounding the MUE transmits signal with P_{low}. Therefore, we propose the SFR pattern selection procedure to cluster FBSs with same pattern. In addition, we consider the use of SON technique in order to avoid increment of complexity by information exchange of FBSs.

Figure 4 shows the procedure of the proposed SFR pattern selection. In the proposed procedure, we assume that all FBSs have DL receiver function and measure received signal strengths (RSSs) in all subbands. When a FBS becomes switched on, the FBS itself measures RSS in each subband by using the DL receiver function. Then, the FBS searches two subbands with maximum RSS (RSS_{max}) and minimum RSS (RSS_{min}), and calculate D_{RSS}, which is the difference between RSS_{max} and RSS_{min} ($D_{RSS} = RSS_{max} - RSS_{min}$), in decibel scale. The FBS determines SFR pattern according to the D_{RSS} value. If D_{RSS} is less than pre-defined threshold D_{RSS_Thr} ($D_{RSS} \leq D_{RSS\_Thr}$), it means that the FBS locates in the border between clusters with different SFR patterns. Therefore, the FBS randomly selects one SFR pattern. On the other hand, if D_{RSS} is more than D_{RSS_Thr} ($D_{RSS} > D_{RSS\_Thr}$), the FBS selects one SFR pattern which has high transmission power in subband with $maxD_{RSS}$ to cluster FBSs with same SFR pattern. After the FBS determines one SFR pattern via the procedure, its transmission power in each subband configures according to the SFR pattern.

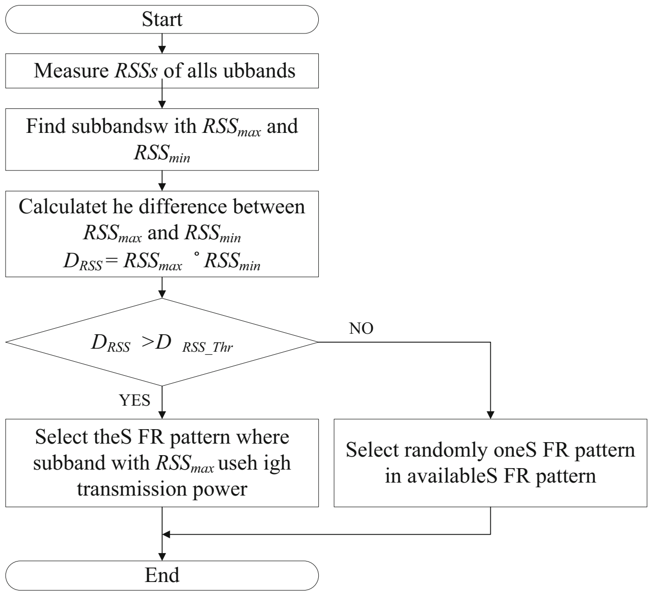


Fig. 4. SFR pattern selection procedure for clustering

4 Performance Evaluation

Performance of the proposed clustering method is evaluated through system level simulations bases on C programming with event driven method. We consider a cellular system with nineteen macrocells with hexagonal shape where one target macrocell locates in the center and surrounded with eighteen interfering macrocells. In the simulation, total 700 FBSs are uniformly deployed in the coverage area of one target macrocell and six neighboring macrocells. MBSs and FBSs have 2GHz carrier frequency and simultaneously use 10MHz system frequency band. We uniformly locate 30 MUEs in coverage area of target MBS and one FUE per FBS in coverage area of a FBS. In order to estimate the channel condition in macrocell and femtocell, we consider the WINNER II path loss models introduced in [9]. We use the urban macrocell model (C1 NLOS) and the indoor model (B3 indoor hotspot) for estimating outdoor and indoor path loss. We also use the indoor to outdoor model (A2 NLOS) to evaluate cross-tier interference between FBS and MUE in downink. We use simulation parameters shown in Table 4 [10][11].

In the considered system environment, we evaluate the cumulative relative frequencies of throughput per UE and the cell throughput of macrocell and femtocell in downlink according to cell planning scheme and method of SFR pattern selection. Figure 5 shows cumulative relative frequencies of throughput

Table 1. Simulation parameters

Parameter	Assumption	
	Macrocell BS	Femtocell BS
Deployment	19 (hexagonal grid)	100 per macrocell (uniform)
Total Tx power	46 dBm	23 dBm
Radius of cell	500m	10m
Number of UEs	30 per cell	1 per cell
Channel model (WINNER II)	C1 NLOS	B3 indoor hotspot
Carrier freq./ Bandwidth	2 GHz / 10 MHz	
Scheduling	PF in frequency and time	
CQI delay	1 ms	
D_{RSS_Thr}	3 dB	

per MUE. When ρ is one, FBSs utilize whole system band with same transmission power (reuse 1). In the channel condition, throughputs for about 10% MUEs are less than 1 kbps due to the interference from aggressor FBSs. In the same channel condition, if aggressor FBSs utilize SFR and randomly select SFR pattern, percentages of MUEs having less than 1 kbps throughputs are reduced to 8% when ρ is 10. On the other hand, if aggressor FBSs utilize the proposed selection scheme, percentages of MUEs having less than 1 kbps throughputs are more reduced to 6% with same ρ value. If ρ become very large, FBS with SFR is similar to that of FBS with reuse 3. While percentages of MUEs having less than 1 kbps throughputs converge to about 0.8% in case of random selection (RS), it are more decreased to about 0.2% when FBS with SFR utilize the proposed SFR pattern selection scheme. From the result, we can find that SFR scheme with proposed scheme can be useful for mitigating interference from aggressor FBS to victim MUE in downlink.

Figure 6 shows cumulative relative frequencies of throughput per FUE according to ρ and method of SFR pattern selection. When ρ is low, throughput of FUE is similar to that of FUE served by FBS with reuse 1. However, when ρ become large, percentages of FUE with less than 10 Mbps converge to about 1.5% and 2%, in case of FBSs with RS and proposed scheme, respectively. In proposed scheme, since FBSs with same SFR pattern make group, interference between FBSs with proposed scheme is relatively high than that of FBS with RS. Nevertheless, since 98% FUEs has still high data rate compared with that of MUEs, it can bear to decrease the performance of FUE.

In Figure 7, we estimate cell throughput of macrocell and femtocell according to ρ and method of SFR pattern selection. In downlink, the cell throughput means that sum of throughput of UEs connected to a BS. When FBSs are not deployed in macrocell, average cell throughput of a MBS is about 15 Mbps. If

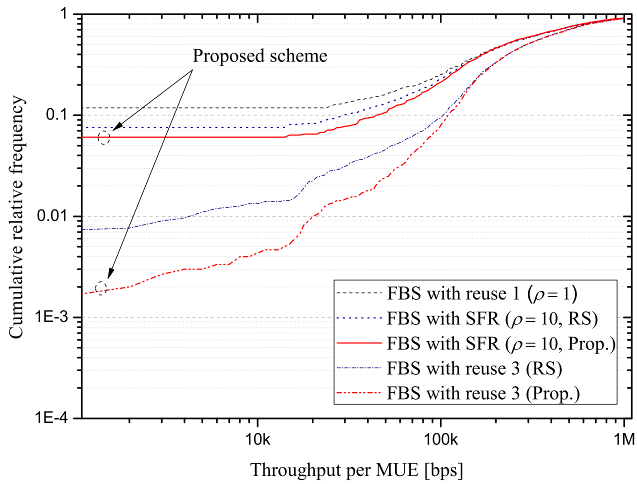


Fig. 5. Cumulative relative frequency of throughput per MUE

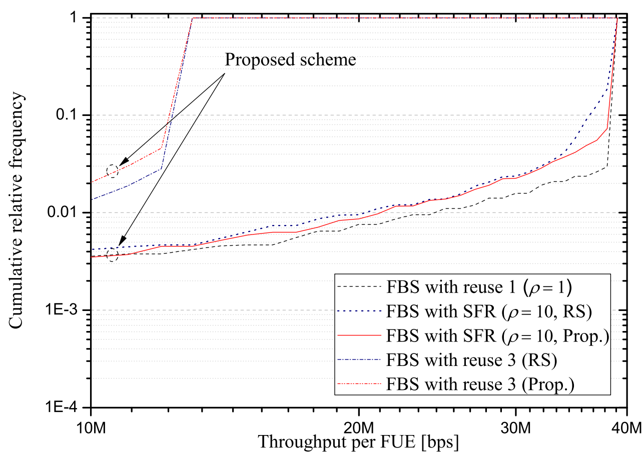


Fig. 6. Cumulative relative frequency of throughput per FUE

FBSs are deployed without proposed scheme, the average cell throughput of a MUE drops into about 11 Mbps due to cross-tier interference by the FBSs. In SFR scheme with $\rho = 10$, we can see that the average cell throughput of a MBS and a FBS are not changed significantly. However, when the ρ increases, average cell throughput of a MBS converges to about 12.5 Mbps and average cell throughput of a FBS is similar to that of a MBS. From the result, we can find the trade off between cell throughput of a MBS and a FBS that is required for optimal value of ρ.

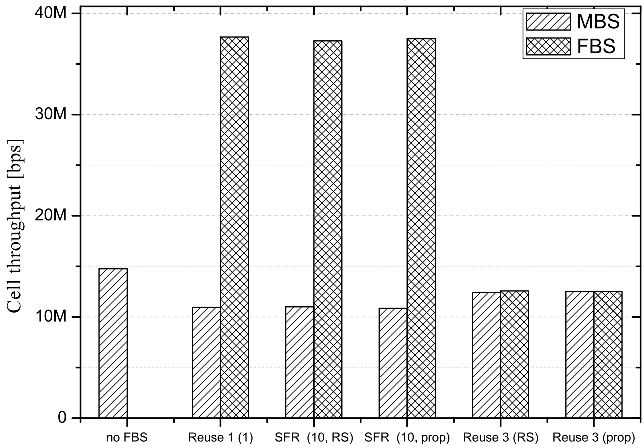


Fig. 7. Cell throughput of each cell

5 Conclusion

In this paper, we adopted SFR scheme to FBS in order to increase efficiency of frequency reuse and mitigate cross-tier interference. In addition, we proposed a SFR pattern selection method which clustering cluster FBSs with same SFR scheme in order to optimize performance of MUEs. For reducing complexity, FBSs listen to the RSS from other FBSs and clusters together by using proposed procedure without exchanging information between FBSs. By the system level simulation, we confirmed that SFR-based FBSs using proposed scheme improve the performance of MUEs clustering FBS with same SFR pattern.

References

1. 3GPP TR 36.814 v9.0.0: Evolved Universal Terrestrial Radio Access (E-UTRA); Further advancements for E-UTRA physical layer aspects (2010)
2. Claussen, H., Calin, D.: Macrocell offloading benefits in joint macro-and femtocell deployments. In: 20th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Tokyo, pp. 350–354 (2009)
3. Lopez-Perez, D., Valcarce, A., de la Roche, G., Jie, Z.: OFDMA Femtocells: A Roadmap on Interference Avoidance. IEEE Communications Magazine 47(9), 41–48 (2009)
4. Lopez-Perez, D., Ladanyi, A., Juttner, A., Zhang, J.: OFDMA Femtocells: Intracell Handover for Interference and Handover Mitigation in Two-Tier Networks. In: IEEE Wireless Communications and Networking Conference, Sydney, pp. 1–6 (2010)
5. Lee, T., Yoon, J., Lee, S., Shin, J.: Interference management in OFDMA Femtocell systems using Fractional Frequency Reuse. In: International Conference on Communications, Circuits and Systems, Chengdu, pp. 176–180 (2010)

6. Chandrasekhar, V., Andrews, J.: Spectrum allocation in tiered cellular networks. IEEE Transactions on Communications 57(10), 3059–3068 (2009)
7. 3GPP TSG-RAN WG4 #52bis: Utility Messages for HeNB ICIC. Qualcomm Europe (2009)
8. 3GPP TR 36.921 v9.0.0: Evolved Universal Terrestrial Radio Access (E-UTRA); FDD Home eNode B (HeNB) Radio Frequency (RF) requirement analysis (2010)
9. IST-4-027756 WINNER II D1.1.2 v1.1: WINNER II Part 1 Channel Models (2007)
10. 3GPP TR 25.814 v7.1.0: Physical layer aspect for evolved Universal Terrestrial Radio Access (UTRA) (2010)
11. 3GPP TSG-RAN WG1 #53bis: Evaluation methodology for LTE-A - Heterogeneous networks. Qualcomm Europe (2008)

A Measurement Based Method for Estimating the Number of Contending Stations in IEEE 802.11 WLAN under Erroneous Channel Condition*

Jun Suk Kim, Bum-Gon Choi, and Min Young Chung**

School of Information and Communication Engineering, Sungkyunkwan University, 300, Chunchun-dong, Jangan-gu, Suwon, Gyeonggi-do, 440-746, Republic of Korea
{jsk7016,gonace,mychung}@ece.skku.ac.kr

Abstract. Performance on throughput of IEEE 802.11 distributed coordination function (DCF) is severely affected by the number of contending stations. Thus, many methods for estimating the number of contending stations have been proposed in order to improve the performance of IEEE 802.11 by adjusting the medium access control (MAC) parameters according to the number of contending stations. However, since the methods estimate the number of contending stations only under the strict assumptions for specific conditions, they cannot correctly estimate the number of contending stations in the general environments. Therefore, in order to estimate the number of contending stations in the general environments, a new estimation algorithm is required. In this paper, we propose a measurement based method for estimating the number of contending stations. From simulation results, we confirm that the proposed method can more accurately estimate the number of contending stations than the existing methods under ideal and erroneous channel conditions.

Keywords: IEEE 802.11, DCF, WLAN, Estimator, Contending stations.

1 Introduction

IEEE 802.11 wireless local area network (WLAN) standard has defined two medium access control (MAC) mechanisms; distributed coordination function (DCF) as a basic channel access mechanism and point coordination function (PCF) as an optional channel access mechanism [1]. DCF is a competitive medium access technique which is used for most of the WLAN stations. PCF is a noncompetitive and centralized medium access control technique. However,

* This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0016757).

** Corresponding author.

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 22–31, 2011.

PCF has been rarely used for WLAN stations because of its high complexity and inefficient operational procedure.

In IEEE 802.11 DCF, a station utilizes binary exponential backoff (BEB) algorithm in order to avoid frame collision when it accesses wireless medium [1]. A station having packets to transmit initializes backoff stage and chooses backoff counter. The backoff counter is randomly chosen at the range of zero to W, where W is the current contention window size. A minimum value of W is CW_{min} and a maximum value is CW_{max}. If a station fails to transmit a packet, backoff stage increases by one and contention window is doubled until the contention window size reaches to CW_{max}.

Fig. 1 shows an example of operations for three stations with IEEE 802.11 DCF capability. After selecting backoff counter, each station observes whether channel is occupied by other stations or not during distributed inter-frame space (DIFS) duration. After the stations determine that the channel is not occupied by other stations during the DIFS duration, all stations having frames to transmit decrease their backoff counter by one whenever the channel is idle during a slot time. If the channel is used by other stations, the stations watching the channel stop decreasing backoff counter. When backoff counter becomes zero, the station tries to transmit their data packets. If a data packet is successfully transmitted, the station initializes backoff stage by zero and waits for next packet transmission. If the packet transmission is failed, the station increases backoff stage by one and attempts the packet transmission by selecting backoff counter from zero to doubled contention window size. When the number of attempts that a station tries to transmit a data packet reaches at retry limit defined in IEEE 802.11 standard, the station discards the data packet, initializes backoff stage, and waits for next packet transmission.

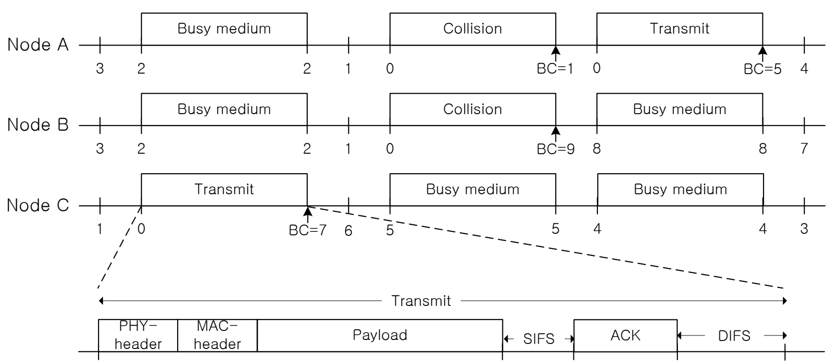


Fig. 1. Basic access method of IEEE 802.11 DCF

A lot of papers have studied the performance of IEEE 802.11 DCF via mathematical analysis. Bianchi proposed a mathematical model for IEEE 802.11 DCF assuming that each station attempts to transmit a packet until the packet is successfully transmitted in the saturation condition [2], [3]. Chatzimisios *et al.*

proposed a similar mathematical model restricting the number of transmission attempts in the saturation condition [4]. The model proposed by Chatzimisios considers not only packet collision but also channel error to analyze DCF performance. In this method, if a station fails packet transmission and its current backoff stage is maximum backoff stage, the station discards the data packet and is ready for next packet transmission. According to these studies, the performance of IEEE 802.11 DCF is very sensitive to the number of stations contending for the data transmission [2]–[6].

To improve the performance of IEEE 802.11 DCF, many methods considering the number of contending stations are proposed based on mathematical models [7]–[9]. Actually, mathematical models are good for analyzing the performance of the network. However, when network parameters are changed, the results of mathematical analysis may be different from the performance of actual networks. These differences between actual and estimated numbers of contending stations increase when the channel bit error rate (BER) or the actual number of contending stations increases. In order to reflect the variation of parameters and channel conditions, the mathematical models should be redesigned. However, it is difficult to design mathematical model considering variety of system parameters and channel conditions, to estimate the number of contending stations. Therefore, a measurement based estimation algorithm for contending stations is required to avoid influence of varying network parameters.

In this paper, we propose the measurement based method for estimating the number of contending stations. In the proposed method, we measure the transmission failure and packet collision probabilities, and estimate the packet error rate by using these two probabilities. Additionally, we measure transmission probability and estimate the number of contending stations based on these probabilities. The rest of this paper is organized as follows. In section 2, we briefly explain the method for estimating the number of contending stations considering transmission errors under erroneous channel condition. The measurement based method for estimating the number of stations is proposed in section 3. Section 4 verifies and analyzes the performance of the measurement based method for estimating the number of contending stations at erroneous and saturated channel condition. Finally we conclude our topic in section 5.

2 Method for Estimating the Number of Contending Stations

There are many researches on mathematical models and analysis utilizing Markov chain model in order to evaluate the performance of IEEE 802.11 DCF [2]–[4]. Bianchi proposed a mathematical model for evaluating DCF performance under ideal channel condition [2], [3]. In this model, a station repeatedly attempts to transmit a data packet until the transmission is successfully completed. In [2], [3], Bianchi proposed a mathematical relation between the probability of transmission failure caused by the packet collision (p_c), the probability attempting data transmission in a randomly selected time slot (τ), and the number of contending stations to access wireless medium for data transmissions (n) as follows:

$$p_c = 1 - (1 - \tau)^{n-1}. \tag{1}$$

On the basis of this mathematical relation, Bianchi proposed a method for estimating the number of contending stations under ideal channel condition [7]. In this method, the number of contending stations can be calculated as a function of p_c and τ as follows.

$$n = 1 + \frac{\log(1 - p_c)}{\log(1 - \tau)}. \tag{2}$$

If all stations in the network know p_c and τ, the stations can independently estimate the number of contending stations by using (2). Each station measures p_c by sensing the channel slots and mathematically calculates τ by using the function of p_c induced by analyzing the Markov chain model proposed in [3].

To analyze the performance of IEEE 802.11 DCF for considering transmission errors of wireless medium, Chatzimisios *et al.* proposed a mathematical model where the transmission is restricted within retry limits [4]. In this model, Chatzimisios *et al.* assumed that stations discard a packet and are ready to transmit next packet when packet transmission is failed at the maximum backoff stage. The probability of transmission failure (p) caused by packet collision or transmission error proposed in [4] is expressed as follows:

$$p = 1 - (1 - \tau)^{n-1}(1 - BER)^{l+H}, \tag{3}$$

or can be expressed as:

$$p = 1 - (1 - \tau)^{n-1}(1 - PER), \tag{4}$$

where BER and PER mean bit error rate and packet error rate, respectively. l and H are the length of payload and header.

From (4), we can represent the number of contending stations under erroneous channel condition as a function of p, τ, and PER:

$$n = 1 + \frac{\log(1 - p) - \log(1 - PER)}{\log(1 - \tau)}. \tag{5}$$

If knowing p, PER, and τ under erroneous channel condition, each station can independently estimate the number of contending stations. Stations measure p by observing the channel slots and mathematically calculate τ by using the function of p induced by analyzing the Markov chain models proposed in [4]. PER is calculated by utilizing p and p_c [10].

In order to evaluate the performance of IEEE 802.11 DCF, a lot of mathematical models were proposed [2]–[4]. These analysis models are used to simply calculate τ for estimating the number of contending stations in the strict condition. However, they cannot effectively calculate τ in the general environments where network parameters varies, as a result, stations cannot correctly estimate the number of contending stations. Therefore, it is necessary for a method accurately measuring τ regardless of the various network parameters.

3 Measurement Based Method for Estimating the Number of Contending Stations

To calculate the probability of transmission failure, each station separately counts the number of transmission attempts and transmission failures. Then, p is calculated as the proportion of transmission failures (N_f) to total transmission attempts (N_t) as follows:

$$p = \frac{N_f}{N_t}. \tag{6}$$

In general, the expression of p in (6) is acceptable in common sense. If the station transmitting a packet does not receive acknowledgment (ACK) message from the destination, the station judges that the packet transmission fails.

The collision probability is calculated as the proportion of busy slots to the sum of idle and busy slots under erroneous channel condition. The time duration where no station transmits packets is called idle slot. On the other hand, the time duration where at least one of the stations transmits a packet is called busy slot. This means that the collision takes place if a station transmits data at the time duration where other stations are transmitting. The stations obtain the collision probability by counting the numbers of idle and busy slots. Based on the counted slots, p_c is calculated as follows [10]:

$$p_c = \frac{N_B}{N_I + N_B}, \tag{7}$$

where N_I and N_B are the numbers of idle slots and busy slots, respectively.

The probability that a data packet is transmitted with channel errors (PER) is calculated by utilizing p and p_c [10]. The relation of PER, p, and p_c is expressed as follows:

$$1 - p = (1 - p_c)(1 - PER). \tag{8}$$

From (8), PER is expressed as follows:

$$PER = 1 - \frac{1 - p}{1 - p_c}, \tag{9}$$

where p is obtained from (6).

A station calculates PER by using p and p_c obtained from (6) and (7), respectively. However, the method for calculating τ is redesigned when system parameter (for example, retry limit) is changed. Therefore, if stations measure τ by sensing the wireless medium without mathematical models in order to avoid influence of the varying network parameters, stations can estimate the number of contending stations by using (5). In this paper, we propose a effective method for measuring τ without mathematical models in order to estimate the number of contending stations under ideal and erroneous channel conditions.

The probability that a station tries to transmit a packet at the randomly selected time slot is notated as the proportion of the number of transmitting slots to the sum of contending (idle in saturated condition) and transmitting slots. τ is expressed as follows:

$$\tau = \frac{N_T}{N_I + N_T}, \tag{10}$$

where N_I and N_T are the number of idle and transmission slots, respectively. This is same as the probability selecting one slot within the total slots where a station is competing and transmitting. Each station counts N_T and N_I through sensing the wireless medium and calculates τ by using these counted slots. (10) also can be expressed as the average of the number of idle slots to one transmission slot as follows:

$$\tau = \frac{1}{\overline{N_I} + 1}. \tag{11}$$

$\overline{N_I}$ means the average number of idle slots. The form of (11) is advantageous for the frequently varying contention window size. Therefore, in this situation, methods for computing $\overline{N_I}$ are necessary to estimate the number of contending stations.

$\overline{N_I}$ can be calculated by using two methods; periodic average and moving average methods. Periodic average method calculates values, (p, p_c, τ, PER, n, etc.), at the end of the defined period. The longer defined period is, the more accurate calculated value is. However, the periodic average method cannot effectively reflect the sudden variation of input values in the period. In order to trace transient inputs, moving average method is computed from the weighted sum of the averages and sampled values at the subsets of full dataset. The average number of idle slots ($\overline{N_I}$) applied with moving average method is calculated through the following formula [7]:

$$\overline{G}(t+1) = \alpha \cdot \overline{G}(t) + (1-\alpha) \cdot G(t), \tag{12}$$

where $\overline{G}(t)$ is the average from the start to tth set of data, $G(t)$ is average value sampled at the tth set of data, and α is weighted factor to trace the actual value in real time.

4 Performance Evaluation

In order to evaluate the performance of proposed estimation method, we consider basic service set (BSS) networks assuming that all stations exist within their transmission range to exclude the effect of hidden node problem. Also the network is assumed as saturation condition that all stations in the network always have data packets to transmit. The system parameter used in DCF is given at Table 1 which refers IEEE 802.11 standard [1]. The lengths of packet header and payload are assumed to be uniformly fixed and the data rate also is supposed to the constant bit rate (CBR). Simulations are carried out by utilizing Monte-Carlo's event-driven method [11].

Table 1. Simulation parameters

Parameters	values
PHY header	128 bits
MAC header	272 bits
Payload	8000 bits
ACK	PHY header + 112 bits
Data rate	54 Mbps
Propagation delay	1 μs
DIFS	34 μs
SIFS	16 μs
Slot time	9 μs
CW_{min}	16
CW_{max}	1024

Simulations are based on (5), (6), (9), (7), and (11) to reflect the erroneous channel condition. In this simulation, a measurement based method for estimating the number of contending stations using auto regressive moving average (ARMA) filter will be verified whether applied in the erroneous channel condition. Simulations are preformed during 200 seconds time duration and about four cases of channel conditions; Without BER, $BER = 10^{-6}$, $BER = 10^{-5}$, and $BER = 10^{-4}$. In these simulations, the number of contending stations is predicted every 10 slots, and 2,000 samples are averaged to plot one point in the figures. Through the simulation, we show that proposed method based on τ measurement can estimate the number of contending stations under ideal even though erroneous channel conditions.

Fig. 2 shows comparison of the estimated numbers of contending stations by utilizing a Markov chain model and τ measurement method. In the simulations, the value of retry limit and BER are set as 7 and as 10^{-4}, respectively. In the Fig. 2, conventional methods based on mathematical models have gaps between input and output values. This gap is caused because Markov chain models do not reflect the variation of retransmit limit. Therefore, we can notice that the proposed method can more accurately estimate the number of stations than conventional mathematical methods when retry limit of system parameter is varying.

Fig. 3 shows differences between input and the estimated number of contending stations. In this simulation, we assume that BER is as 10^{-4} and retry limit changes from 6 to 8 by one. In the mathematical model based method, the gaps between the number of actual contending stations and the estimated number of stations are growing when the number of actual stations are increasing or the retry limits are decreasing. This result comes from the stationary Markov chain model used in mathematical analysis. As a result, mathematical method cannot accurately estimate the number of contending stations under condition of transient retry limits. On the other hand, we can notice that the τ measurement based estimation method can estimate it even though under condition of

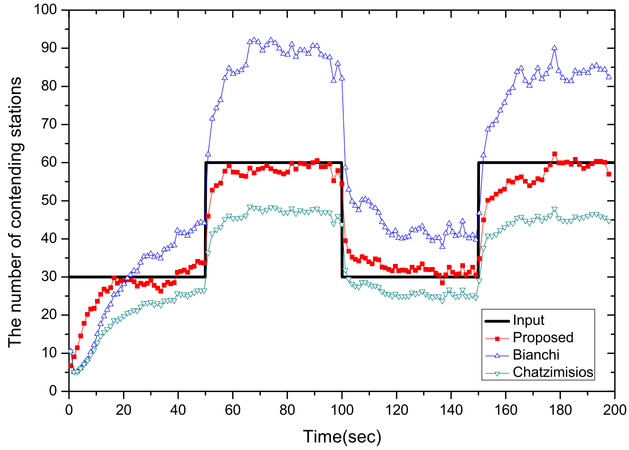


Fig. 2. The comparison between modeling based estimations and measurement based estimation

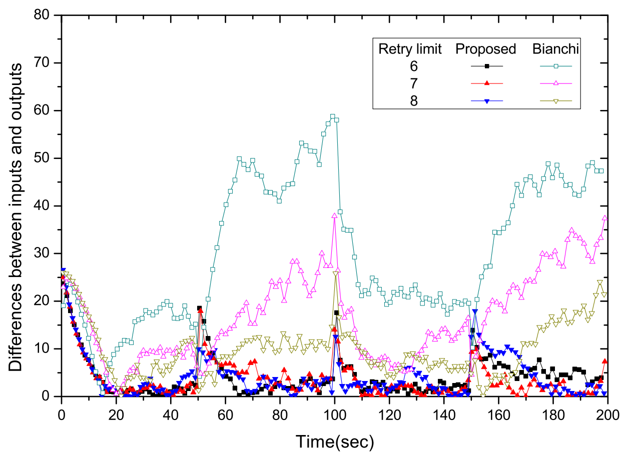


Fig. 3. Errors between actual stations and estimated values

transient retry limits because τ measurement is not affected by varying system parameters.

In Fig. 4, we can see the estimated number of contending stations to the variation of weighted vector α. There is the comparison of estimation results from $\alpha = 0.995$ and $\alpha = 0.999$. The closer α is to 1, the more accurate the estimating the number of contending stations is. However, the trace velocity of input values is more slowly. Also Fig. 5 shows results of estimations in the typical channel condition which is without BER, $BER = 10^{-6}$, $BER = 10^{-5}$, and $BER = 10^{-4}$. The measurement based estimation can be applied adaptively in erroneous channel condition even if transmission errors are increasing. Through the Figs. 2–5, we can see that the station can estimate the number of actual stations

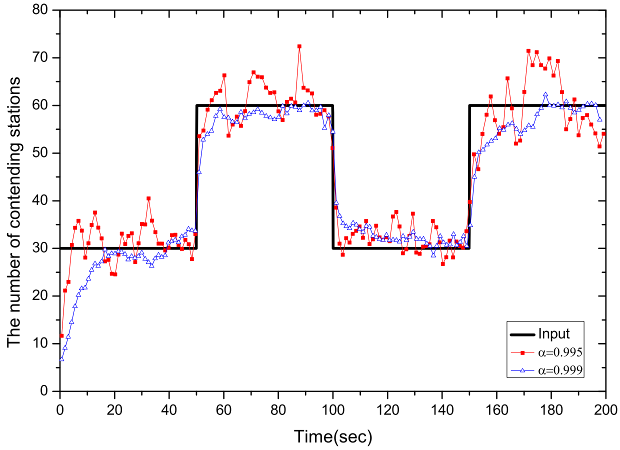


Fig. 4. τ measurement based estimation with weighted factors

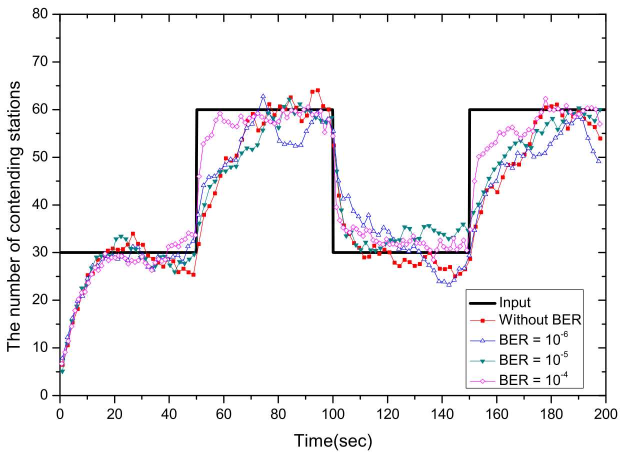


Fig. 5. τ measurement based estimation in various channel condition

by using τ measurement based estimation method to use only information obtained through channel sensing and this proposed method can be applied even though erroneous channel condition.

5 Conclusion

In this paper, we proposed the τ measurement based method for estimating the number of contending stations in IEEE 802.11 WLAN under ideal and erroneous channel conditions. The base for the proposed method is provided by the effective expression form of τ given as a relation between the numbers of idle and transmission slots. Proposed method enables to directly measure τ with sensing the wireless medium and solve the problem of mathematical analysis having high reliance to system parameters. The simulation results show that proposed method can effectively estimate the number of contending stations with variations of retry limit under ideal and erroneous channel conditions. And proposed method can estimate it by using suitable α which affects to both trace velocity and accuracy of estimation. Our work will offer an accurate and effective solution for estimating the number of contending stations in the general environments. Therefore, we assure that our proposed method contributes many related works for improvement of IEEE 802.11 DCF performance.

References

1. IEEE: Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Standard 802.11 (2007)
2. Bianchi, G.: IEEE 802.11 – Saturation Throughput Analysis. IEEE Communications Letters 2(12), 318–320 (1998)
3. Bianchi, G.: Performance Analysis of IEEE 802.11 Distributed Coordination Function. IEEE Journal on Selected Areas in Communications 18(3), 535–547 (2000)
4. Chatzimisios, P., Boucouvalas, A.C., Vitas, V.: Performance Analysis of IEEE 802.11 DCF in Presence of Transmission Errors. In: IEEE International Conference on Communications (ICC), vol. 7, pp. 3854–3858 (2004)
5. Wang, C., Li, B., Li, L.: A New Collision Resolution Mechanism to Enhance the Performance of IEEE 802.11 DCF. IEEE Transactions on Vehicular Technology 53(4), 1235–1246 (2004)
6. Tay, Y.C., Chue, K.C.: A Capacity Analysis for the IEEE 802.11 MAC Protocol. ACM/Baltzer Wireless Networks 7(2), 159–171 (2001)
7. Bianchi, G., Tinnirello, I.: Kalman Filter Estimation of the Number of Competing Terminals in an IEEE 802.11 Network. In: IEEE International Conference on Computer Communications 2, pp. 844–852 (2003)
8. Kim, J.S., Serpedin, E., Shin, D.R.: Improved Particle Filtering-Based Estimation of the Number of Competing Stations in IEEE 802.11 Networks. IEEE Signal Processing Letters 15, 87–90 (2008)
9. Vercauteren, T., Toledo, A.L., Wang, X.: Batch and Sequential Bayesian Estimators of the Number of Active Terminals in an IEEE 802.11 Network. IEEE Transactions on Signal Processing 55(2), 437–450 (2007)
10. Malone, D., Clifford, P., Leith, D.J.: MAC Layer Channel Quality Measurement in 802.11. IEEE Communications Letters 11(2), 143–145 (2007)
11. Hammersley, J.M., Handscomb, D.C.: Monte–Carlo Methods. Methuen & Co. LTD., New York (1964)

Implementation of WLAN Connection Management Schemes in Heterogeneous Network Environments with WLAN and Cellular Networks*

Hyung Wook Cho, Sueng Jae Bae,
Hyunseung Choo, and Min Young Chung**

School of Information and Communication Engineering
Sungkyunkwan University
300, Chunchun-dong, Jangan-gu, Suwon, Gyeonggi-do, 440-746, Republic of Korea
{chohw,noooi,choo,mychung}@ece.skku.ac.kr

Abstract. As increasing demands for mobile communications, importance of utilizing wireless local area network (WLAN) rapidly increases in order to assist cellular networks and to reduce communication cost. In this paper, we propose a WLAN connection management scheme which helps mobile node (MN) to appropriately access WLANs considering minimum required data rates of ongoing services. Moreover, in order to evaluate performance of proposed and conventional connection management schemes, we construct testbed consisting of WLANs and 3GPP high speed downlink packet access (HSDPA) network. In addition, we implement server and client applications on MN having both WLAN and HSDPA interfaces. From test results, we find that the proposed connection management scheme can provide steadier and higher data rate than the conventional connection management scheme adopted in Google Android operating system (OS).

Keywords: Heterogeneous networks, QoS, RRM, Testbed implementation, WLAN.

1 Introduction

Recently, many attractive mobile devices, such as smart phones, tablet devices, and ultra mobile personal computer (UMPC), have been introduced, and they are rapidly propagated in the global market. Since those mobile devices cover various mobile communication services, e.g., Internet, voice over IP (VoIP), video on demand (VoD), on-line games, e-mail, and so on, demands for mobile data

* This research was supported by Future-based Technology Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (No. 2010-0020727).
** Min Young Chung is corresponding author.

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 32–43, 2011.

communications have been extremely increased. For example, in AT&T networks, wireless data traffic has increased by 5000% over the past three years because of the propagation of mobile devices [1].

Due to the increased demands for wireless data communications, wireless service providers are required to expand capacity of their networks in order to satisfy customers' demands and guarantee quality of service (QoS). However, it is difficult to suddenly increase the capacity of current wireless communication networks because of high cost and complexity. In order to increase the capacity of wireless communication networks with minor changes and low cost, many researches on cellular networks and wireless local area network (WLAN) internetworking have been performed in [2]-[6]. In general, WLAN and cellular networks have opposite characteristics. IEEE 802.11 WLAN operates in industrial, scientific and medical (ISM) radio bands and they can provide higher data rate to users than cellular networks [2]. However, service coverage of WLAN is limited in small area (hotspot zone) and mobility between different access points (APs) is not supported in IEEE 802.11 WLAN. Contrary, cellular networks use licensed radio band and it covers larger area and support higher mobility than WLAN. However, due to its large coverage and limited radio resources, cellular networks have lower data rate and high communication cost than WLAN. Because of these complementary characteristics, both users and wireless service providers can take advantage of those networks if we can effectively integrate WLAN and cellular networks [3]-[7].

Especially, cellular networks can accommodate more users with high data rates by sharing its traffic loads with WLANs. However, since some schemes in previous researches are too complex and they do not consider practical limitations of WLAN interface, they cannot be directly adopted in real network environments. Thus, not only efficient scheme is required but also intensive tests in real networks are required to enhance performance of cellular networks and WLANs internetworking.

In this paper, we propose a new WLAN connection management scheme based on the received signal strength (RSS) and required data rate of ongoing services in heterogeneous network environments consisting of cellular networks and WLANs. In addition, we build testbed including WLANs, 3GPP high speed downlink packet access (HSDPA) network, server and client applications, and mobile node having both WLAN and HSDPA interfaces, and evaluate performance of the proposed connection management scheme. By using the testbed, we compare performance of the proposed connection management scheme with that of the conventional connection management scheme used in general smart phones [8][9].

The rest of this paper is organized as follows. Section 2 describes radio resource management in WLAN and linux configuration application programming interface (API) for wireless interface. Our proposed WLAN connection management scheme is introduced in Section 3. In Section 4, we introduce a testbed for performance evaluation and analyze performance of our proposed scheme in Section 5. Finally, Section 6 gives the conclusion.

2 Related Works

2.1 Radio Resource Management in WLAN

Recent wireless cellular systems, such as long-term evolution (LTE) of 3rd generation partnership project (3GPP) and IEEE 802.16m of IEEE, include various radio resource management (RRM) schemes and functionalities in order to efficiently utilize licensed radio bands, since they should guarantee QoS for their customers [10][11]. On the other hand, in IEEE 802.11 WLAN, RRM is rarely considered because conventional WLANs are expected to provide best-effort services to users. However, in cellular network and WLAN internetworking, IEEE 802.11 WLAN is required to consider RRM because service providers should ensure transparency to cellular customers using WLAN.

RRM functionalities in IEEE 802.11 WLAN can be facilitated in both access point (AP) and mobile node (MN) as shown in Fig. 1 [12]. RRM functions in WLAN APs can be categorized by radio frequency (RF) grouping, dynamic channel assignment, transmit power control, and coverage hole detection and correction algorithms. In RF grouping algorithm, WLAN controller (WLC) manages RF group, which is a group of neighboring APs using same RF band, in order to facilitate other RRM functionalities (dynamic channel assignment, transmit power control, and coverage hole detection and correction). Dynamic channel assignment and transmit power control (TPC) algorithms deal with assignment of RF bands and adjustment of transmit powers of WLAN APs for load management and interference mitigation, respectively. Coverage hole detection algorithm assists TPC algorithm by detecting coverage hole based on the signal quality of MNs, while coverage hole correction algorithm suggests to WLC that some APs are required to increase their transmit power to remove coverage holes.

RRM functions in MN can be categorized into connection management and AP selection algorithms. Connection management algorithm manages wireless connections of an MN by switching connection between cellular network and WLAN and by deciding connection switching timing based on signal quality and/or QoS measurement of ongoing services. In AP selection algorithm, MN decides appropriate AP which can provide stable connection and guarantee QoS among neighboring APs. In addition, by considering loads of the neighboring APs, AP selection algorithm can disperse loads of APs and increase performance of overall WLANs. In this paper, we concentrate on the WLAN connection management algorithm in order to guarantee QoS for ongoing services.

2.2 Wireless Tool for Linux

In order to implement various RRM techniques in MN with IEEE 802.11 WLAN interface, some parts of mobile system, e.g., kernel input/output (I/O) controller and I/O control functions in operating system (OS), wireless interface drivers, and software with RRM functionalities, should be modified. In this paper, we use and modify wireless tools in Linux system to implement the proposed and conventional WLAN control management algorithms.

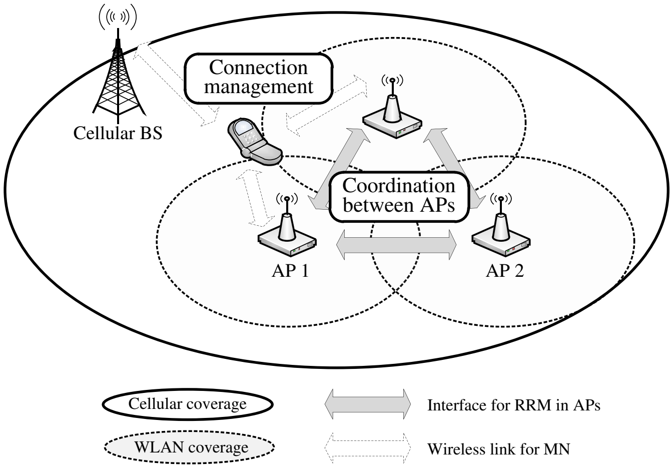


Fig. 1. RRM in AP and MN

Wireless tools are developed by Hewlett-Packard (HP) since 1996 in order to manage, configure, and facilitate wireless interfaces, especially IEEE 802.11 WLAN. They use wireless extension (WE) application programming interface (API) in Linux system which allows WLAN interface drivers to expose configuration and statistics specific information on WLAN interface to the user space [8].

Wireless tools consist of five package tools, i.e., *iwconfig*, *iwlist*, *iwspy*, *iwpriv*, and *ifrename*. *iwconfig* allows user program to manipulate basic wireless parameters, such as frequency band, transmit power, and AP address. By using *iwlist*, user program can initiate WLAN scanning and list basic information on discovered APs. *iwspy* and *iwpriv* help user programs to acquire link qualities of neighboring nodes and to manipulate the WE specific configurations of a WLAN driver, respectively. *ifrename* makes user program to rename WLAN interface based on various static criteria. In this paper, we modify *iwconfig* and *iwlist* wireless tool packages to discover WLANs, change attached AP, and manage connection of MN in the testbed.

3 Proposed Scheme

General OSs for mobile systems adopt simple WLAN connection management algorithm due to the cost and complexity. Let's consider network environments shown in Fig. 2 and an MN which moves from (A) to (F). In the connection management scheme in Google Android OS [9], the MN maintains connection with AP 1 until it crosses the boundary of AP 1's coverage ((C) in Fig. 2) even though the MN communicates with low data rate due to the weak received signal strength (RSS). After escaping the coverage of AP 1, the MN performs WLAN scanning while it continues communications through cellular BS. When the MN discovers an AP included in its white list ((D) in Fig. 2), the MN immediately establishes connection with the AP. However, since the RSS from AP 2 is low at the position (D), the MN may suffer from low data rate.

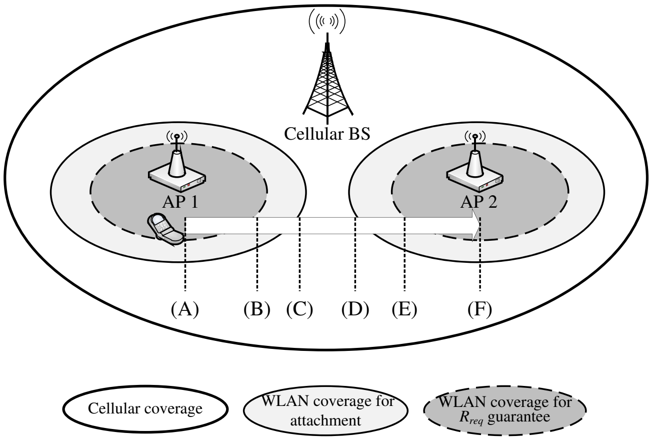


Fig. 2. An example of network environments with cellular network and WLANs

In order to resolve the problem, we propose a WLAN connection management scheme which can efficiently utilize WLANs and guarantee minimum required data rate ($R_{required}$) for ongoing communication services. For the proposed connection management scheme, we consider a MN having wireless interfaces for both cellular network and WLAN. We assume that the cellular network interface is always turned on and it maintains connection to the cellular BS for cellular communication services, e.g., voice call and messaging services. In addition, we assume that the MN prefer WLAN than cellular network, since WLAN is expected not only to provide higher data rate with lower communication cost than cellular networks but also to reduce loads of cellular BSs.

Let's assume that an MN moves from (A) to (F) in Fig. 2 and currently communicates with AP 1 using WLAN interface. The MN periodically calculates average received signal strength (RSS) ($\bar{S}_{WLAN}(\Delta T \cdot k)$) and SNR (signal to noise ratio) threshold margin ($Th(\Delta T \cdot k)$) with period ΔT. To minimize the effect of multi-path fading and shadowing, $\bar{S}_{WLAN}(\Delta T \cdot k)$ (dBm) is calculated as an average of RSS measurement samples as follows:

$$\bar{S}_{WLAN}(\Delta T \cdot k) = \frac{1}{N} \sum_{n=0}^{N-1} S_{WLAN}(\Delta T \cdot k - \frac{\Delta T}{N}), \tag{1}$$

where $S_{WLAN}(t)$ is RSS measured at time t and N is the number of measurements during ΔT. Since the MN should measure RSSs from AP with period

$\Delta T/N$ for the calculation of $\bar{S}_{WLAN}(\Delta T \cdot k)$, ΔT and N is decided by considering capability of WLAN interface and wireless channel environments.

If we assume that the RSS linearly decreases during $2\Delta T$, average RSS at next period $(\Delta T \cdot (k+1))$ can be estimated as

$$\begin{aligned}\tilde{S}_{WLAN}(\Delta T \cdot (k+1)) &= \bar{S}_{WLAN}(\Delta T \cdot k) \\ &\quad - \left\{\bar{S}_{WLAN}(\Delta T \cdot (k-1)) - \bar{S}_{WLAN}(\Delta T \cdot k)\right\} \\ &= 2 \cdot \bar{S}_{WLAN}(\Delta T \cdot k) - \bar{S}_{WLAN}(\Delta T \cdot (k-1)). \qquad (2)\end{aligned}$$

If $\tilde{S}_{WLAN}(\Delta T \cdot (k+1))$ becomes too low, data rate of WLAN may decrease and $R_{required}$ cannot be guaranteed at next period. Thus, we define $Th(\Delta T \cdot k)$ as

$$Th(\Delta T \cdot k) = \alpha \cdot MCS(\tilde{S}_{WLAN}(\Delta T \cdot (k+1))), \qquad (3)$$

where $MCS(x)$ is achievable data rate when RSS is x dBm according to the modulation and coding scheme (MCS) table in IEEE 802.11 standard [2] and α $(0 < \alpha \leq 1)$ is a weight which adjusts ideal data rate to real environments considering point coordination function (PCF) contention, noise, overhead, and so on. If $R_{required} \leq Th(\Delta T \cdot k)$ as the result of kth examination, the MN keeps current connection with the AP since the AP is expected to guarantee $R_{required}$ until next examination (during (A)-(B) in Fig. 2). On the other hand, if $R_{required} > Th(\Delta T \cdot k)$ (at (B) in Fig. 2), the MN considers that currently attached AP cannot guarantee $R_{required}$ until next period and it switches its connection to cellular networks. After switching connection to cellular network, the MN periodically performs WLAN scanning. If the MN finds an AP which satisfies $R_{required} \leq Th(\Delta T \cdot k)$ (at (E) in Fig. 2), it establishes new connection with the AP and uses the AP for communications.

4 Testbed Implementation

In order to evaluate performance of proposed and conventional connection management schemes, we have implemented testbed consisting of WLAN and cellular network. Fig. 3 shows network architecture of the testbed, which consists of two WLANs and one cellular network. To implement WLANs, two APs (IPTIME N604R) with IEEE 802.11n capability and one Ethernet switch (3COM Office Connect, model no. 3C16792A) is used [13][14]. The switch is connected to router (Cisco Catalyst 6509) which provides connection to Internet through 500Mbps connection link provided by Korea telecommunications (KT) [15]. For cellular communications, Node B (NB), which is a base station of HSDPA networks, and universal mobile telecommunications system (UMTS) provided by SK Telecommunications (SKT) is used.

MN in the testbed is implemented in laptop PC (Samsung R70). For wireless communications, the laptop PC is equipped with both WLAN and cellular interfaces. Linksys WUSB54G WLAN interface is used for WLAN communications. It includes RT2526 and RT2571 chipsets which are 2.4 GHz transceiver and IEEE 802.11b/g medium access control (MAC) / base band processor (BBP),

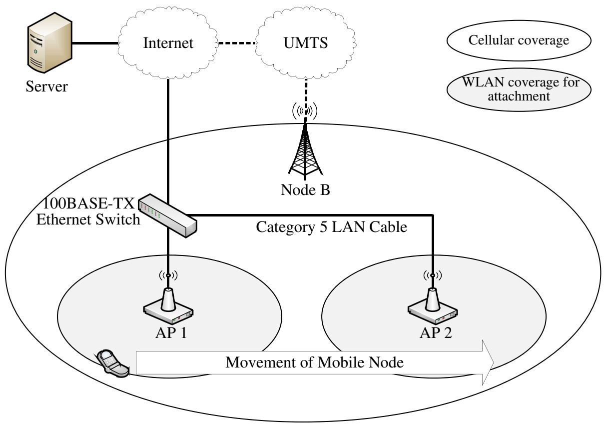


Fig. 3. Network Architecture of Testbed

respectively[16]. For cellular communications, we use cellular interface (IM-H100 of Pantech Inc.) based on 3GPP HSDPA standard (release 5, category 3). The cellular interface is connected to laptop PC by universal serial bus (USB) and it provides data rates up to 1.8 Mbps and 384 Kbps in downlink and uplink communications, respectively.

In order to apply the proposed and conventional access control schemes introduced in Section 3, we modified wireless tools and implemented connection management modules in Linux system (Ubuntu 10.10 with Linux kernel 2.6.35). Fig. 4 shows placement of the connection management modules. The connection management modules measure link quality and controls WLAN and cellular connections by utilizing *iwconfig* and *iwlist* wireless tools. In addition, we write simple server and client applications which communicate each other and measure performance metrics based on C and socket programmings. Server application is located in server PC and it generates constant bit rate (CBR) traffic destinated to MN. Since server and client applications are synchronized, client application can measure average data rate, frame loss ratio, delay, and delay variance.

5 Performance Evaluation

For performance evaluation, we conduct intensive experiments using the testbed explained in Section 4. Experiments were performed on the 2nd floor of 1st engineering building in Sungkyunkwan University, Korea, as shown in Fig. 5 obtained by utilizing google map [17]. Node B for HSDPA is approximately 500 meters apart from the test area and signal quality from the Node B is sufficient for full rate HSDPA communications. Two APs are installed at each end of hallway and distance between the APs is 65 meters. RSS characteristics for the test environments are shown in Fig. 6. The server and client applications use transmit control protocol/Internet protocol (TCP/IP) since the UMTS and HSDPA networks limit unauthorized user datagram (UDP) communications. In

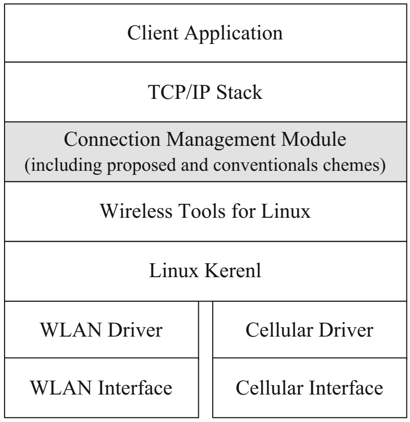
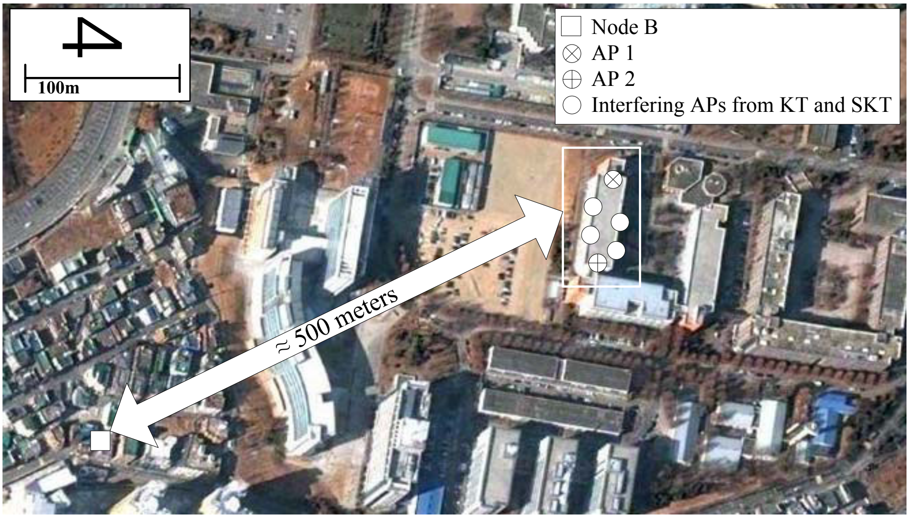


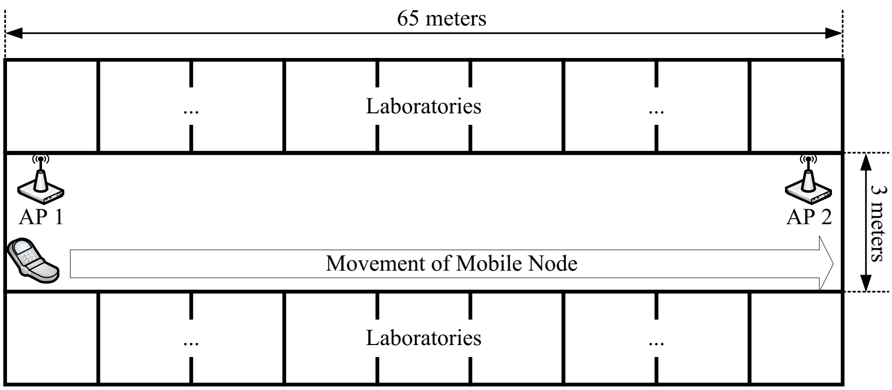
Fig. 4. Connection module in Linux system

addition, we do not consider mobility in IP since we focus on link connectivity in the paper. For experiments, we set WLAN bandwidth and AP transmission power to 20 MHz and 20 dBm, respectively. MN can keep a connection to WLAN AP whose RSS is larger than -82 dBm. We assume that velocity of MN is 0.5 m/s and $R_{required}$ is 700 Kbps. In order to accurately measure RSS, ΔT and N are set to five seconds and 20 samples, respectively. In addition, we set α to 0.03 based on repetitive measurements.

In order to analyze WLAN and cellular connectivity of MN, we evaluate data rates for conventional and proposed connection management schemes in Fig. 7. Data rates are measured at the client application in MN, and it is averaged by using moving window with three samples window size in order to improve visibility of the figure. At the start of test, MNs for both conventional and proposed schemes are attached to AP 1 and communicate with server application through the AP. Since distances between the MNs and AP 1 are close, data rates for both schemes satisfy $R_{required}$ for 0 to 24 seconds. However, as the distances between MNs and AP 1 increase, data rates fluctuate because of low RSS and large path loss. As shown in the figure, in spite of the poor link quality, MN with conventional connection management scheme keeps connection with AP 1 until it escapes coverage area of AP 1 ((C1) in Fig. 7). After escaping the coverage area of AP 1, the MN immediately establishes connection with AP 2 as the result of WLAN scanning procedure. However, $R_{required}$ (700 Kbps) still cannot be guaranteed until 123 seconds because link quality for AP 2 is low. After 123 seconds, link quality for AP 2 increases, and the MN tries to perform delayed data transmissions from server by using TCP mechanism. Because of that, data rate in the conventional scheme rapidly increases (around 130 seconds in Fig. 7). On the other hand, MN with proposed connection management scheme switches its connection from AP 1 (WLAN) to Node B (HSDPA) when it considers that AP 1 cannot guarantee $R_{required}$ based on RSS from AP 1 ((P1) in Fig. 7).



(a) Experiment environments



(b) Magnified part (2nd floor of 1st engineering building in Sungkyunkwan University)

Fig. 5. Experiment environments for proposed and conventional connection management schemes

As the result, the MN can continuously receive data through cellular network (Node B) although data rate is temporarily decreased during 60-70 seconds due to connection establishment and scheduling processes in HSDPA networks. In addition, although AP 2 is discovered, the MN postpones connection establishment to the AP 2 until the AP can provide data rate more than $R_{required}$. When the MN with proposed connection management scheme considers that the AP 2 can guarantee $R_{required}$ ((P2) in Fig 7), it establishes connection with the AP and switches its connection from cellular network to WLAN.

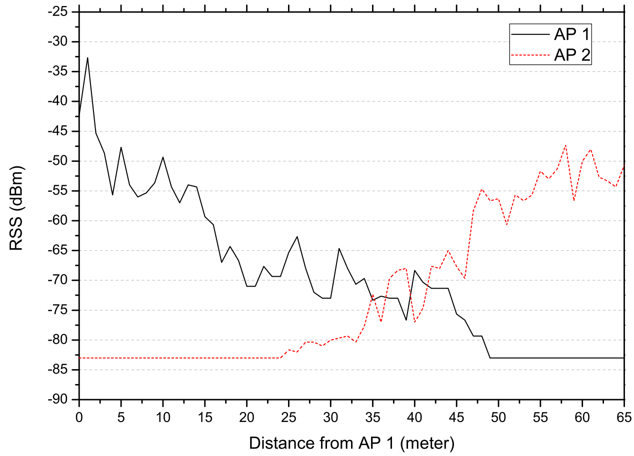


Fig. 6. RSS characteristics for the test environments

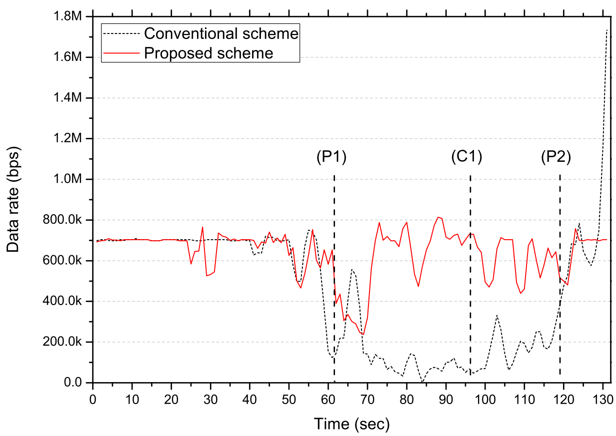


Fig. 7. Data rates for proposed and conventional connection management schemes

In Fig. 8, we evaluate accumulated data volume in bits for conventional and proposed connection management schemes to analyze effect of the connection managements. For conventional connection management scheme, data volume is rarely increased after (P1) due to the poor link quality between MN and AP 1. On the other hand, for proposed connection management scheme, accumulated data volume continuously increase without rapid change of increasing ratio since the proposed scheme effectively manages wireless connections between WLANs and cellular networks. As the results of these reasons, accumulated data volume for the proposed connection management scheme is approximately 14 percent larger than that for the conventional connection management scheme.

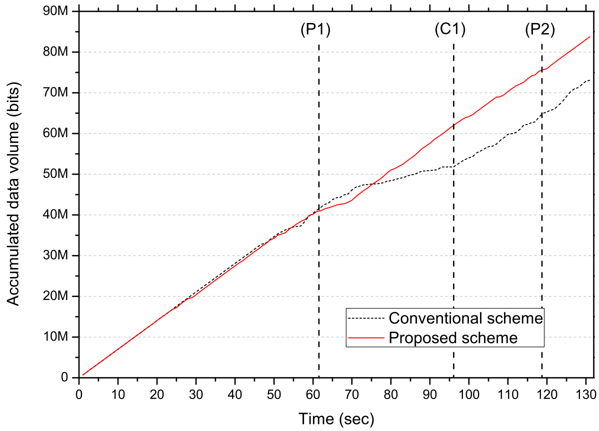


Fig. 8. Accumulated data volumes for proposed and conventional connection management schemes

6 Conclusion

In this paper, we have proposed a new connection management scheme which decides attachment and detachment for WLAN APs in order to guarantee minimum required data rate of ongoing services. In addition, we have built a testbed consisting WLAN, HSDPA network, server, and mobile node with dual-mode capability which enables tests on mobile communication techniques, and have performed intensive tests for the proposed and conventional connection management schemes. From the test results, we found that the steady data rate can be provided by adopting connection management scheme. Thus, we conclude that the proposed connection management scheme can help both service provider and customers to guarantee QoS with low cost and complexity and to be experience consistent QoS, respectively.

For further study, we will enhance drivers for WLAN and cellular network interfaces in order to provide seamless wireless service provision. In addition, testbed will be extended to support mobile IP and IEEE 802.21 media independent handover (MIH) which are core technologies for seamless mobile communications.

References

1. AT&T (February 2010), `http://www.att.com`
2. IEEE Std. IEEE Standard Local and metropolitan area networks, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE (June 2007)
3. Bernaschi, M., Cacace, F., Iannello, G., Za, S., Pescape, A.: Seamless Internetworking of WLANs and Cellular Networks: Architecture and Performance Issues in a Mobile IPv6 Scenario. IEEE Wireless Communications 12(3), 73–80 (2005)

4. Salkintzis, A.K., Fors, C., Pazhyannur, R.: WLAN-GPRS Integration for Next-Generation Mobile Data Networks. IEEE Wireless Communications 9(5), 112–124 (2002)
5. Chou, C.M., Huang, C.Y.: Dynamic vertical handover control algorithm for WLAN and UMTS. In: IEEE Wireless Communications and Networking Conference (WCNC), pp. 606–610 (2006)
6. Pahlavan, K., Krishnamurthy, P., Hatami, A., Ylianttila, M., Makela, J.P., Pichna, R., Vallstron, J.: Handoff in Hybrid Mobile Data Networks. IEEE Pers. Commun. 7(2), 34–47 (2000)
7. Smadi, M., Azhari, V.S., Todd, T.D., Kezys, V.: A Study of WLAN-to-Cellular Handover Using Measured Building-Exit Data. IEEE Transaction on Vehicular Technology 58(4), 2044–2054 (2009)
8. Hwelett-Packard: Wireless Tools for Linux, http://www.hpl.hp.com
9. Google Open Source Project: Android 2.2 (Froyo), http://source.android.com
10. 3GPP TS 36.331 Ver. 9.3.0: Radio Resource Control (RRC) Protocol Specification (2010)
11. IEEE 802.16m-07/002r2: Draft IEEE 802.16m Requirement (2006)
12. Cisco Systems: Technical Notes: Radio Resource Management under Unified Wireless Networks (2010)
13. Iptime, http://www.iptime.co.kr
14. 3com, http://www.3com.com/products/
15. Cisco, http://www.cisco.com
16. Linksys, http://www.linksysbycisco.com
17. Gogle map, http://maps.google.com

Wireless Sensor Network's Lifetime Maximization Problem in Case of Given Set of Covers*

Adil I. Erzin[1,2] and Roman V. Plotnikov[2]

[1] Sobolev Institute of Mathematics,
Siberian Branch of the Russian Academy of Sciences, Novosibirsk, Russia
[2] Novosibirsk State University, Novosibirsk, Russia

Abstract. We consider a wireless sensor network's lifetime maximization problem as an integer linear programming problem when the redundant set of covers is given, and it is necessary to choose a lifetime of each cover subject to the limited sensor's resources. Irrespective of the function of any sensor in a cover, we suppose that the sensor may be active during some time, and the number of time rounds when it is active we call the sensor's resource. We proved that the considered problem is strong NP-hard; proposed the ways to reduce the problem; shown that the problem is not approximable within the 0.997; found special cases when it is in the class APX; proposed several heuristics and performed a simulation.

Keywords: Wireless sensor network, lifetime maximization, energy consumption.

1 Introduction

Sensor is an autonomous intellectual device which is responsible for sensing, data processing, receiving and transmitting. It may be in active mode performing its functions and consuming energy and in a sleep mode when the energy consumption is negligible. In the wireless sensor network (WSN) each sensor has a limited non-renewable energy (resource). In some cases one can measure resource in time rounds during which the sensor may be active. Since the number of sensors in WSN considerably exceeds the minimum necessary amount, then the subsets of sensors may perform the function of WSN. One of the most important issues in WSN is energy consumption optimization, whence it follows the WSN's lifetime maximization.

Usually a *cover* is defined as a subset of sensors which cover the required region or the given set of targets. In most problems the covers are required

* This research was supported jointly by the Russian Foundation for Basic Research (grant 10-07-92650-IND-a) and by Federal Target "Grant Scientific and educational personnel of innovation Russia" for 2009-2013 (government contract No. 14.740.11.0362).

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 44–57, 2011.

to have certain properties (connectivity [1], special structure [2], etc). In [3] and [4] sensing ranges are adjustable and therefore energy consumption of an active sensor by one unit of time may differ depending on particular cover. In [5], the authors formulate the WSN's lifetime maximization problem in case of continuous variables as a packing linear program and propose a heuristics based on the Garg-Konemann algorithm [6]. In this paper we assume that the redundant set of covers is given. Since every sensor can consume any amount of its energy (it depends specifically on sensing and communication ranges), we suppose that the sensor's energy is measured in the time rounds. This assumption is not too severe, as after initializing the sensor, its parameters, as a rule, do not change over time, and therefore, in any cover sensor consumes the same amount of energy during one time round.

In section 2 we give basic definitions and formulate the WSN's lifetime maximization problem as an integer linear programming (ILP) problem. In section 3 we prove NP-hardness of the problem. In section 4 several ways to simplify the problem in some special cases are proposed. In section 5 the approximability of the problem is analyzed. We proved that the problem does not have a polynomial-time approximation scheme (PTAS) and found some additional conditions when it belongs to the APX class. In section 6 we consider some special cases when the problem may be solved by polynomial time complexity. In section 7 two approximation algorithms are proposed. We have implemented all designed algorithms and present the simulation results in section 7. Section 8 concludes the paper.

2 Problem Formulation

Let, for example, O be the set of targets, J be the set of sensors, $|J| = n$. Let $r_j \in Z^+$ be the initial resource of sensor $j \in J$, i.e. the number of time rounds during which the sensor may be active. We suppose that the set of different covers $C = \{C_1, \dots, C_m\}$, $C_i \subseteq J$, is given. Let parameter $b_{jk} = 1$ if sensor $j \in C_k$ and $b_{jk} = 0$ otherwise. We use the variable vector $y = (y_1, \dots, y_m)$ where $y_k \in Z^+$ is the lifetime of cover C_k (the number of time rounds when the sensors in C_k are active). Then the WSN's lifetime maximization problem is as follows.

$$\sum_{k=1}^{m} y_k \to \max_{y_k \in Z_0^+} \; ; \tag{1}$$

$$\sum_{k=1}^{m} b_{jk} y_k \le r_j, \;\; j \in J \; . \tag{2}$$

For every cover $C_k \in C$ let us define its *resource* r^k as a minimal resource of sensors in C_k. Obviously, r^k is the upper bound for the lifetime of the cover C_k. Denote the minimal initial resource of all sensors as $r_{\min}$, the maximal - as $r_{\max}$.

A sensor is *critical* if its resource is less than the sum of resources of all covers which contain it.

3 NP-Hardness

In case of continuous variables the problem (1)-(2) is a linear programming problem and therefore may be solved by polynomial time. Let's prove that in the case of integer variables this problem is NP-hard.

Consider the special case of the problem (1)-(2) (when the resources of sensors are ones).

$$\sum_{k=1}^{m} y_k \to \max_{y_k \in \{0,1\}} ; \tag{3}$$

$$\sum_{k=1}^{m} b_{jk} y_k \leq 1, \; j \in J . \tag{4}$$

To formulate the next lemma remind a well-known maximal independent set problem (MIS): find the maximal cardinality independent subset of vertices in a given graph.

Lemma 1. *Problems (3)-(4) and MIS are equivalent.*

Proof. Solution to the problem (3)-(4) is the maximal cardinality set of *disjoint* covers. Construct the intersection graph as follows. Each vertex of this graph corresponds to the one cover. Two vertices are adjacent iff the corresponding covers have at least one common sensor. Notice that the problem (3)-(4) is reduced to MIS by constructing a covers intersection graph. It is obvious that such constructing may be done by polynomial time.

In order to prove the opposite polynomial reduction, let us consider an arbitrary MIS instance in some graph $G = (V, E)$. The problem is to find a max-cardinality independent subset of nodes in V. Evidently any isolated vertex is assuredly included in the optimal solution, so it can be excluded. Below the polynomial procedure reduces MIS to (3)-(4).

Let $V_k = \{i \in V | i > k, \; (k, j) \in E\}$. In order to construct the sets $C_k \subset V$, $k = 1, \ldots, m$ we propose the following procedure.

```
begin
  for all i ∈ {1,...,m} do C_i := ∅;
  k := 1;
  l := 0;
  while V_k ≠ ∅ or k ≠ m do
  begin
    if V_k ≠ ∅ then
    begin
      i := min{j ∈ V_k};
```

```
      l := l + 1;
      C_k := C_k ∪ {l};
      C_i := C_i ∪ {l};
      V_k := V_k \ {i};
    end
    else k := k + 1;
  end
end
```

The procedure defines parameters $\{b_{jk}\}$: $b_{jk} = 1$ if $j \in C_k$ and $b_{jk} = 0$ otherwise. These parameters define an instance of the problem (3)-(4). So, this procedure is a polynomial reduction of MIS to the problem (3)-(4) with coincidence of objective functions. □

Corollary 1. *The problem (1)-(2) is strong NP-hard.*

Proof. It is known that MIS is strong NP-hard [7]. In Lemma 1 we reduced MIS to the instance of (1)-(2) by polynomial time. Therefore (1)-(2) is also strong NP-hard. □

Although the evidence of NP-hardness of the problem is quite natural, we could not find it in the literature

4 Simplifications

In certain cases WSN may contain sensors which may be excluded from J. Sometimes the set of covers C may be reduced without decreasing of WSN's lifetime. Let us consider several special cases below.

1. The cover C_i contains the cover C_j as a subset. Obviously the cover C_i is not advantageous because while it is active all resources of sensors from C_j are consumed. Besides, resources of all sensors from $C_i \setminus C_j$ are consumed as well. Therefore, the cover C_i may be excluded from C.
   We say, vector $a = (a_1, \dots, a_n)$ *dominates* vector $b = (b_1, \dots, b_n)$ if for all $i \in \{1, \dots, n\}$ $a_i \leq b_i$. The above covers elimination is equivalent to the deletion of dominated columns from the constraint matrix.
2. Two sensors belong to the same covers. In this case their resources are always consumed simultaneously. One may leave only one sensor with the less value of resource, another sensor may be excluded from J. Moreover, if k sensors belong to the same set of covers, one may keep only one of them with the less value of resource.
3. Any non-critical sensor may be excluded from J. If sensor belongs to more than one cover and its resource is not less than the sum of resources of covers which contain it, then this sensor may be excluded.

After the implementation of above simplifications, all sensors which are included in more than one cover will be critical, the constraint matrix $B = ||b_{ij}||$ will contain only different rows and no column will dominate another.

5 Approximability

Let us fix parameters a, b and δ: $0 < a \leq b$, $\delta \geq 3$. Consider the following problem $P_{a,b,\delta}$:

$$\sum_{k=1}^{m} y_k \to \max_{y_k \in Z_0^+} ;$$

$$\sum_{k=1}^{m} b_{jk} y_k \leq r_j,\ j \in \{1, \ldots, n\} ;$$

$$a \leq r_j \leq b,\ j \in \{1, \ldots, n\} ;$$

$$\sum_{l=1,\ l \neq k}^{m} \mathrm{sgn}(\sum_{j=1}^{n} b_{jl} b_{jk}) \leq \delta,\ k \in \{1, \ldots, m\} .$$

Here the resource of every sensor belongs to the fixed segment $[a, b]$, and every cover intersects with at most δ other covers.

Lemma 2. *The problem $P_{a,b,\delta}$ belongs to the class APX.*

Proof. Consider MIS in graph with degree δ, and denote such problem as MIS-δ. It is known that this problem belongs to the class APX. That is, for every fixed $\varepsilon > 0$ there is such polynomial algorithm A_ε that the following inequality is true for any instance of MIS-δ [8]

$$\frac{F_\varepsilon}{F^*} \geq \frac{5}{\delta + 3 + \varepsilon} , \tag{5}$$

where F_ε is cardinality of independent set yielded by the algorithm A_ε, and F^* is the optimal objective value. For any instance $I \in P_{a,b,\delta}$ consider a covers intersection graph G. Its degree is not more than δ. Let $\varepsilon > 0$. Apply algorithm A_ε to the problem in the graph G. Let $\widetilde{F}_\varepsilon$ be the cardinality of independent set constructed by the algorithm A_ε, let $\widetilde{F}^*$ be the cardinality of maximal independent set in the graph G. According to (5) the following inequality is true:

$$\frac{\widetilde{F}_\varepsilon}{\widetilde{F}^*} \geq \frac{5}{\delta + 3 + \varepsilon} .$$

Algorithm A_ε finds disjoint covers and, therefore, it constructs a feasible solution of I. Let W_ε be the lifetime of WSN defined by the built solution, let W^* be the maximal sensor network's lifetime. A lifetime of the cover in the solution is equal to the cover resource. Therefore, $W_\varepsilon \geq r_{\min} \widetilde{F}_\varepsilon$. Let k be the chromatic number of G. Then $\widetilde{F}^* \geq \frac{m}{k}$. It is known [9], that $k \leq \delta + 1$. Therefore, $\widetilde{F}^* \geq \frac{m}{\delta+1}$. So, $W^* \leq r_{\max} m$ entails $W^* \leq (\delta + 1) r_{\max} \widetilde{F}^*$. Finally,

$$\frac{W_\varepsilon}{W^*} \geq \frac{r_{\min} \widetilde{F}_\varepsilon}{(\delta + 1) r_{\max} \widetilde{F}^*} \geq \frac{5a}{(\delta + 1)(\delta + 3 + \varepsilon) b} .$$

Since a, b and δ are constants, then $P_{a,b,\delta} \in$ APX. $\square$

Corollary 2. *Any general problem which consists of finite number of instances of (1)-(2) belongs to class APX.*

Lemma 3. *The problem (1)-(2) is not approximable within 0.997.*

Proof. It is known that for any $\delta > 0$ there is such $\varepsilon > 0$ that the construction of $\delta^{-\varepsilon}$-optimal solution for MIP-δ is NP-hard [10]. Since the problem MIP-δ is equivalent to $P_{1,1,\delta}$, the problem $P_{1,1,\delta}$ cannot be approximated within the factor 0.997 [11] either. Therefore, the problem (1)-(2) is not approximable within 0.997. □

6 Special Cases

Let us consider several special cases when the problem (1)-(2) may be solved by polynomial time complexity.

6.1 Total Unimodularity of a Constraint Matrix

Since (1)-(2) is ILP problem, then it may be solved by polynomial time in case when the constraint matrix B is totally unimodular (TU) [12]. Let us find sufficient conditions when B is TU. We use the following criteria of TU.

1. [12] An $n \times m$ integer matrix A is TU iff for any subset of row's indices $R \subset \{1, \dots, n\}$ there exists such its partition into two disjoint sets R_1 and R_2 that
$$|\sum_{j \in R_1} a_{jk} - \sum_{j \in R_2} a_{jk}| \leq 1 \; \forall k \in \{1, \dots m\} \,. \tag{6}$$
2. A matrix A is TU iff a transposed matrix A^T is TU

Lemma 4. *If the set of sensors may be partitioned into two nonempty sets so that none of the covers contains sensors from both sets, then the problem (1)-(2) is polynomially solvable.*

Notice that to fulfill Lemma 4 conditions every cover must include at most 2 sensors. On the other hand, the cover which contains a single sensor may be excluded from C (see section 4). So, in this case one may represent the model as a graph $G' = (J, C)$ with vertices corresponding to sensors and edges corresponding to covers. Let us proof of the following

Lemma 5. *If every cover includes at most 2 sensors and G' is a tree, then solution of the problem (1)-(2) may be obtained efficiently in time $O(n^2)$ by the following algorithm.*

Algorithm A1

```
begin
  (y_1,...,y_m) := (0,...,0);
  S := {C_1,...,C_m};
  while S ≠ ∅ do
  begin
    Find all leaves of G';
    For every found leaf i make LEAF CUTTING (i);
  end
end

LEAF CUTTING (i)
begin
  find vertex j - the parent of the leaf i in G';
  find k: cover C_k corresponds to an edge (i,j);
  y_k := min(r_i, r_j);
  r_j := r_j - y_k;
  S := S \ C_k;
end
```

Proof. The sufficient condition for the optimality of the solution yielded by the above algorithm A1 is the fact that the procedure LEAF CUTTING doesn't impair the solution. Without loss of generality we may assume that the vertex 1 is a leaf of the graph G, the vertex 2 is its parent, C_1 is the cover which corresponds to the edge (1, 2), $\{C_1, C_2, \ldots, C_s\}$ is the set of covers which include the vertex 2. Let vector $(y_1^*, \ldots, y_m^*)$ be the optimal solution to the problem, let $y_1' = \min(r_1, r_2)$. Suppose that the procedure LEAF CUTTING made for the vertex 1 decreases an objective function. In this case for all feasible variables $y_1, ..., y_s$ the next inequality is true: $y_1' + \sum_{y=2}^{s} y_k < \sum_{y=1}^{s} y_k^*$. Sequentially calculate the values $y_2', \ldots, y_s'$ in the following way: $y_l' = \min(y_l^*, r_2 - \sum_{k=1}^{l-1} y_k')$. Since $(y_1^*, y_2^*, \ldots, y_m^*)$ is optimal solution, then the vector $(y_1', \ldots, y_s', y_{s+1}^*, \ldots, y_m^*)$ is feasible. Notice, that

$$\sum_{k=2}^{s} y_k' = \min(\sum_{k=2}^{s} y_k^*, r_2 - y_1') ,$$

therefore,

$$\sum_{k=1}^{s} y_k' = \min(y_1' + \sum_{k=2}^{s} y_k^*, r_2) .$$

Since $y_1^* \le \min(r_1, r_2) = y_1'$ and $r_2 \ge \sum_{k=1}^{s} y_k^*$, then $\sum_{k=1}^{s} y_k' \ge \sum_{k=1}^{s} y_k^*$. It contradicts with above conjecture that the procedure LEAF CUTTING made for the vertex 1 decreases an objective function. Therefore, the optimality of solution $(y_1', \ldots, y_s', y_{s+1}^*, \ldots, y_m^*)$ follows from the optimality of solution $(y_1^*, y_2^*, \ldots, y_m^*)$. □

Lemma 6. *If the covers intersection graph is bipartite, then the problem (1)-(2) is polynomially solvable.*

Proof. If the covers intersection graph is bipartite, then the set of covers C may be partitioned into two nonempty subsets in such way that none of two covers from the same subset contains the common sensors. It means that the condition (6) is true for the matrix B^T. Therefore, the matrix B is TU. □

Since the proof of the next lemma is voluminous, it is omitted.

Lemma 7. *If the covers intersection graph is a tree, and all sensors have the same initial resources, then the solution of the problem (1)-(2) may be found by time $O(m^2)$.*

6.2 Other Special Cases

In this section we briefly describe other special cases when the problem (1)-(2) is solvable by polynomial time.

1. Every cover has a single sensor which is included in other covers. In this case C may be partitioned into the disjoint subsets defined by a common sensor. The problems obtained by such division are *independent*, therefore solution of the entire problem may be yielded by theirs consecutive solving. Notice, that in each subproblem only initial resource of the single critical sensor defines WSN's lifetime and its optimal solution consists of choosing arbitrary non-exhausted cover during every time round while such cover exists.
2. Every cover has common sensors with another single cover. The problem may be decomposed into independent subproblems with at most two covers. We may exclude all subproblems with one cover which are trivial and concentrate on subproblems with two intersecting covers. The optimal solution of the whole problem may be found by time $O(n + m)$ with consecutive solving of subproblems.
3. Some instances of (3)-(4) are solvable by polynomial time. As it was proved in section 3, if resource of every sensor is 1, then the considered problem is equivalent to MIS. It is known [13] that MIS may be solved by polynomial time in the following cases:
   (a) graph G is an edge graph;
   (b) graph G is chordal;
   (c) the graph's degree is 2.
   So, if covers intersection graph of some instance of (3)-(4) satisfies one of above conditions, then the problem (3)-(4) may be solved by polynomial time.

7 Heuristics

In this section we propose the heuristics for the problem (1)-(2) and perform a posteriori analysis.

7.1 Algorithms Description

Let us present the first algorithm H1 below. Initially calculate a *deficit* for every sensor j which is defined as $\sum_{k=1}^{m} b_{jk} - r_j$ and sort the sensors by non-decreasing order of deficits. During one iteration of **while** operator there is performed a run-through over all sensors. Every sensor successively consumes one unit of resource for every cover which contains it. An algorithm stops when resource of every cover is zero. On each iteration of **while** operator the sum of covers resources R is calculated. Since the condition $R > 0$ guarantees an existence of at least one cover with non-zero resource, then R decreases with every iteration of the cycle. Therefore, the number of iterations is at most R.

Algorithm H1

```
begin
  (y_1,...,y_m) := (0,...,0);
  sort sensors in ascending order of deficits;
  for all j ∈ J do V_j := {C_k ∈ C : b_jk = 1};
  sort all elements in V_j in decreasing order of cover resources;
  simplify a problem using the results of section 4;
  R := Σ_{k=1}^m r^k;
  while R > 0 do
  begin
    for all j ∈ J do
      for all C_k ∈ V_j do
        if r^k > 0 then
          begin
            y_k++;
            r^k--;
            for all j_k ∈ C_k do r_{j_k}--;
            for all l ∈ {1,...,m}: C_l ∩ C_k ≠ ∅ do recalculate r^l;
            R := Σ_{l=1}^m r^l;
          end
  end
end
```

Since $O(mn\sum_{k=1}^{m} y_k) \leq O(r_{max}n^2m)$, the algorithm H1 has a pseudopolynomial complexity.

In algorithm H2 the resource of each sensor is divided among the covers which contain it proportionately to the resources of the covers. If equal division is impossible, then the residue of division is distributed among the covers.

Algorithm H2

```
begin
  (y_1,...,y_m) := (0,...,0);
```

```
  sort sensors in ascending order of deficits;
  for all j ∈ J do V_j := {C_k ∈ C : b_jk = 1 };
  sort all elements in V_j in decreasing order of cover resources;
  simplify a problem using the results of section 4;
  for all j ∈ J do
  begin
    calculate S - the sum of resources of covers from V_j;
    for all k ∈ {1,...,m} do p_k := ⌊r_j (r^k/S)⌋;
    calculate the rest of sensor j resource: q := r_j - Σ_{k:C_k∈V_j} p_k;
    for all k ∈ V_j do
      if r^k > 0 then
      begin
        if r^k > p_k then
        begin
          d := sgn(q);
          x := p_k + d;
          q := q - d;
        end
        else
        begin
          x := r^k;
          q := q + (p_k - r^k);
        end
        y_k := y_k + x;
        r^k := r^k - x;
        for all j_k ∈ C_k do r_{j_k} := r_{j_k} - x;
        for all l ∈ {1,...,m}: C_l ∩ C_k ≠ ∅ do recalculate r^l;
      end
  end
end
```

In the worst case (each sensor is included in every cover) this algorithm has complexity $O(n^2 m^2)$, but after simplifications (refer to section 4) one will obtain a trivial case with one sensor and one cover.

7.2 Simulation

All algorithms presented in the previous subsection are implemented in programming language C++. The experiments were run for randomly generated testcases. The area of monitoring was a square region 100m × 100m. The sensor nodes were randomly distributed over the region. For every cover we generated a uniform triangular grid with randomly positioned start point and fixed side of equilateral triangle d. Each cover corresponds to one grid and consists of sensors, which are nearest to the grid nodes. If the distance between the grid node and its nearest sensor is less than d, then we say that the sensor *covers* the grid node. We assume that the region is fully covered if all grid nodes are covered. Obviously a case with small value of d requires large number of sensors. We used six variants of testcases where the number of sensors (n) and d were balanced. The first variant: $d = 35$m and $n = 25$, the second variant: $d = 25$m and $n = 50$, the third variant: $d = 10$m and $n = 100$, the fourth variant: $d = 15$m and $n = 200$, the fifth variant: $d = 10$m and $n = 500$ and the sixth variant: $d = 5$m and $n = 1000$. For each generated testcase we calculate the covered part (CP). In order to evaluate a dependency among covers, for each sensor we calculate a number of covers which contain it (NC). We ran our experiments with resources uniformly assigned (20) to each sensor node, randomly assigned resources between 1 to 5 to each sensor node and randomly assigned resources between 10 to 50 to each sensor node. We use IBM CPLEX system (under the Agreement for IBM Academic Initiative) to obtain an optimal solution for each instance. For $n < 500$ each experiment ran 100 times, for $n \geq 500$ each experiment ran 50 times. The quality of every algorithm was measured as an average value of the ratio F_A/F^*, where F_A is the value of the objective obtained by algorithm, and F^* is the maximum value of objective. So, $F_A/F* \leq 1$, and the bigger the ratio the better solution was yielded by the algorithm A. The average values of NC and CP were also computed for each case.

Table 1. Simulation results

n	m	d	$r_{\min}$	$r_{\max}$	NC	CP (%)	ratio(SR)	ratio(greedy)	ratio(H1)	ratio(H2)
50	10	25	1	5	4.08	99.3	0.832	0.832	0.857	0.857
50	50	25	1	5	18.8	99.5	0.652	0.762	0.751	0.751
50	10	25	20	20	4.07	99.4	0.983	0.949	0.97	0.949
50	50	25	20	20	20.9	99.2	0.905	0.633	0.731	0.655
50	10	25	10	50	4.01	99.7	0.986	0.888	0.87	0.846
50	50	25	10	50	18.9	99.6	0.942	0.782	0.782	0.734
100	10	20	1	5	3.27	100	0.717	0.852	0.827	0.827
100	50	20	1	5	15.1	99.9	0.388	0.753	0.716	0.716
100	10	20	20	20	3.43	99.9	0.948	0.767	0.888	0.767
100	50	20	20	20	15.3	99.9	0.89	0.572	0.75	0.572
100	10	20	10	50	3.23	99.8	0.981	0.838	0.87	0.819
100	50	20	10	50	14.7	99.8	0.934	0.763	0.774	0.714
500	10	10	1	5	2.64	100	0.32	0.87	0.9	0.9
500	50	10	1	5	11.6	100	0.0333	0.687	0.673	0.673
500	10	10	20	20	2.87	99.9	0.908	0.74	0.848	0.74
500	50	10	20	20	12.6	100	0.826	0.597	0.815	0.597
500	10	10	10	50	2.62	100	0.933	0.681	0.866	0.671
500	50	10	10	50	12.5	100	0.827	0.632	0.75	0.608
1000	10	5	20	20	4.81	99	0.988	1	1	1
1000	50	5	20	20	23	99	0.99	1	1	1
1000	10	5	10	50	5.2	98.9	0.957	0.911	0.978	0.911
1000	50	5	10	50	24.1	98.8	0.872	0.838	0.919	0.835

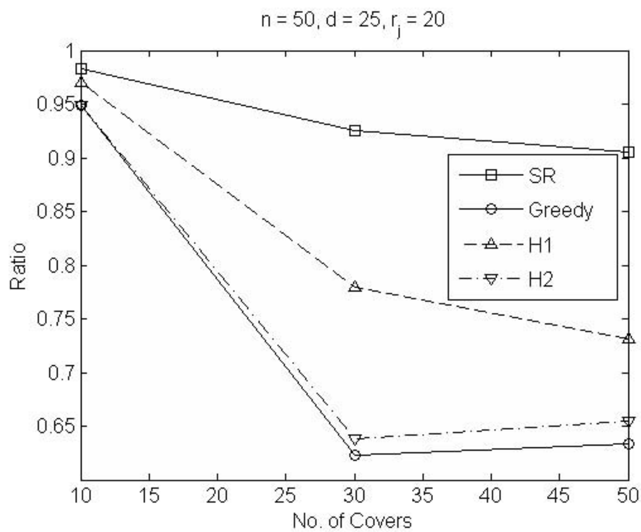


Fig. 1. Greedy algorithm yields the worse result

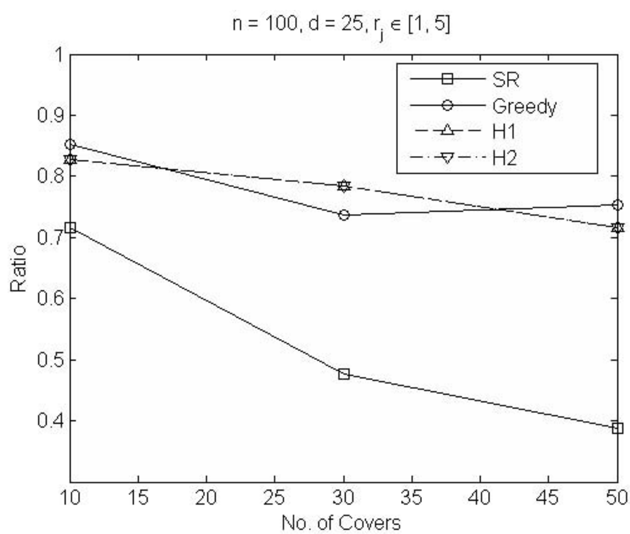


Fig. 2. Algorithms H1 and H2 yield the same results, SR yields the worse result

Besides we compared the results yielded by our algorithms with two trivial approaches. The first one is greedy algorithm, where at each step the cover with maximal resource is selected as active, and then all resources are recalculated. The second one is simplex method plus rounding (SR), i.e taking the integer parts of the optimal continuous variables. In the table above one can see the results of simulation. For example, in most cases algorithm H1 yields better solution than H2 and the greedy algorithm performs worse than the best one

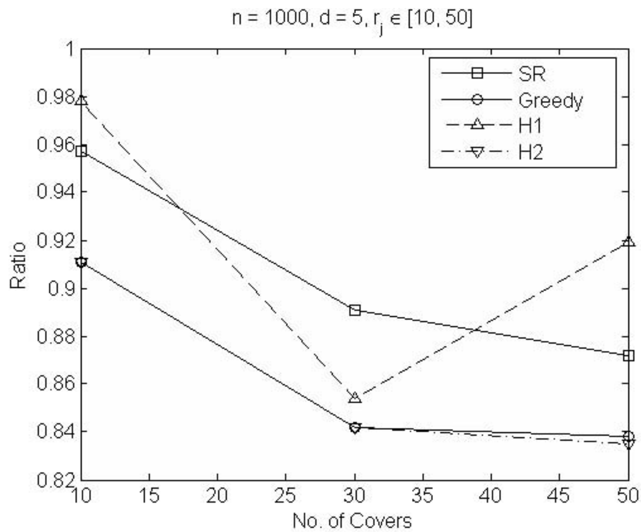


Fig. 3. Greedy algorithm and H2 yield almost the same results, the highest ratio is reached by SR and H1

among the proposed algorithms. Notice that SR has quite good ratio only in the cases when resources are large enough. In the case of small resources the results yielded by SR are rather poor.

In the figures above we illustrate the ratios with respect to the number of covers in some cases. In fig. 1 there is a case with 50 sensors, where resource of each sensor equals 20. In this case both proposed algorithms yield better solution than the greedy algorithm. In fig. 2 one may see a case with 100 sensors, when the resources are small. Notice, that algorithms H1 and H2 yield the same results in this case, their ratios are close to the ratio of greedy algorithm and substantially larger than ratio of SR. Fig. 3 shows a case with 1000 sensors, where resource of each sensor equals 20. In this case all considered algorithms yield high ratio (not less than 0.83), greedy algorithm and H2 almost always yield the same ratio which is less than the ratios of H1 and SR, H1 yields the best solution when the number of covers is 10 and 50.

8 Conclusion

In this paper we have considered the WSN's lifetime maximization problem, when the redundant set of covers is given. Although it remains polynomially solvable for the case of continues variables, we proved that in the case of integer variables it is strong NP-hard. We have also established that the problem does not have PTAS and found some additional conditions when it belongs to the class APX. We proved that the problem is polynomially solvable in some special cases. We proposed two heuristics for solving this problem. We experimentally examined the quality of the proposed algorithms comparing them with the greedy algorithm and simplex method plus rounding. To obtain an optimal solution to the problem we used the IBM CPLEX system, which was granted under the Agreement for IBM Academic Initiative.

In this work it was assumed that a set of covers is given. In further research we plan to develop efficient methods for constructing the set of covers. At the same time the sensor's energy consumption, dependent on the sensing area and on the transmission distance can be taken into account.

References

1. Inanc, M., Magdon-Ismail, M., Yener, B.: Power Optimal Connectivity and Coverage in Wireless Sensor Networks. Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY (2003)
2. Segal, M.: Improving Lifetime of Wireless Sensor Networks. Network Protocols and Algorithms 1(2) (2009) ISSN 1943-3581 2009
3. Cardei, M., Wu, J., Lu, M., Pervaiz, M.O.: Maximum Network Lifetime in Wireless Sensor Networks with Adjustable Sensing Ranges. In: Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob., vol. 3, pp. 438–445 (2005)
4. Zalubovsky, V., Astrakov, S., Erzin, A., Choo, H.: Energy-efficient Area Coverage by Sensors with Adjustable Ranges. Sensors 9(4), 2446–2460 (2009)
5. Berman, P., Galinescu, G., Shan, C., Zelikovsky, A.: Power efficient monitoring management in sensor networks. In: Proceedings of IEEE Wireless Communications and Networking Conference, WCNC 2004, Atlanta, USA, March 21-25, pp. 2329–2334 (2004)
6. Garg, N., Konemann, J.: Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In: Proceedings of FOCS (1997)
7. Garey, M.R., Johnson, D.S.: Computers and Intractability. A guide to the theory of NP-completeness. Freeman, New York (1979)
8. Berman, P., Fujito, T.: Approximating independent sets in degree 3 graphs. In: Sack, J.-R., Akl, S.G., Dehne, F., Santoro, N. (eds.) WADS 1995. LNCS, vol. 955, pp. 449–460. Springer, Heidelberg (1995)
9. Harary, F.: Graph Theory. Addison-Wesley, Reading (1969)
10. Alon, N., Feige, U., Wigderson, A., Zuckerman, D.: Derandomized graph products. Computational Complexity 5, 60–75 (1995)
11. Berman, P. Karpinski, M.: On some tighter inapproximability results. Technical Report TR98-065, ECCC (1998)
12. Ghouila-Houri, A.: Caract'erisation des matrices totalement unimodulaires. Comptes Rendus Hebdomadaires des S'eances de l'Acad'emie des Sciences 254, 1192–1194 (1962)
13. Gavril, F.: Algorithms on circular-arc graphs. Networks 4, 357–369 (1974)

An Implementation of SVM-Based Gaze Recognition System Using Advanced Eye Region Detection

Kue-Bum Lee, Dong-Ju Kim and Kwang-Seok Hong

School of Information and Communication Engineering, Sungkyunkwan University,
300, Chunchun-dong, Jangan-gu, Suwon, Kyungki-do, 440-746, Korea
{leo0608,radioguy}@skku.edu, kshong@skku.ac.kr

Abstract. There have been many recent studies on gaze recognition in the field of Human-Computer Interaction (HCI). Gaze recognition and other biomedical signals will be a very natural and intuitive part of Human-Computer Interaction. In studies on gaze recognition, identifying the user is the most applicable task, and it has had a lot of attention from many different studies. Most existing research on gaze recognition has problems with universal use because the process requires a head-mounted infrared Light Emitting Diode (LED) and a camera, both expensive pieces of equipment. Cheaper alternatives like webcams have the disadvantage of poor recognition performance. This paper proposes and implements the Support Vector Machine-based (SVM) gaze recognition system using one webcam and an advanced eye region detection method. In this paper, we detected the face and eye regions using Haar-like features and the AdaBoost learning algorithm. Then, we used a Gabor filter and binarization for advanced eye region detection. We implemented a Principal Component Analysis (PCA) and Difference Image Entropy-based (DIE) gaze recognition system for the performance evaluation of the proposed system. In the experimental results, the proposed system shows 97.81% recognition of 4 directions, 92.97% recognition of 9 directions, demonstrating its effectiveness.

Keywords: Human-Computer Interaction, Gaze Recognition, Support Vector Machine, Gabor filter, Eye Region Detection.

1 Introduction

In recent research on Human-Computer Interaction (HCI), sense recognition technologies have been used in various human-computer interaction applications. In particular, gaze recognition-based interfaces are popular studies for HCI because it can be very useful in the field of human interface research. A typical example is that gaze recognition can substitute for typing on a keyboard and moving a mouse. Disabled people who do not have the full use of their hands can still type using gaze recognition technology [1]. Gaze recognition can also be used to collect information for a web site usability evaluation to observe the perception behaviors of the users [2].

There are two methods for traditional gaze recognition. The first method uses facial movement. The second method uses eye movement.

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 58–69, 2011.

The gaze recognition systems using facial movement attach markers to a specific position in the face, and by tracking the movement of these markers the system keeps track of the user's gaze [3]. Another method tracks the position of facial features such as the eye, nose, and mouth to recognize a user's gaze [4]. These systems have disadvantages if markers or facial features are out of camera range, exhibit poor performance, or cannot detect subtle changes. When gaze recognition systems track pupil movement, the pupil is identified by the glint of the pupil and the corneal reflection. The direction of gaze is recognized by extrapolating the position and shape distortion information of the pupil and corneal reflection [5]. Another method recognize gaze is by using an infrared Light Emitting Diode (LED) and a stereo camera [6]. However, since these systems must be mounted on the head, they lead to user discomfort.

Consequently, we propose and implement a Support Vector Machine-based (SVM) gaze recognition system using an advanced eye region detection method and a webcam. Accurate eye region detection is essential for gaze recognition. We detect the corner of the eye and the iris using a Gabor filter and binarization for accurate eye region detection. Gabor filters have been used extensively as feature extractors for various applications such as facial recognition, facial expression recognition and fingerprint recognition. We use a Gabor filter, not as a feature extractor, but to detect the eye regions.

The proposed gaze recognition system consists of three steps. The first step of gaze recognition is the detection of the face and eye regions in images acquired by a webcam. We utilize the AdaBoost algorithm based on Haar-like features to accomplish this step. The second step is the detection of the eye corner and iris for advanced eye region detection. Finally, the system recognizes the gaze using SVM.

The remainder of this paper is organized as follows. In Section 2, we describe related work. Section 3 describes the proposed method for advanced eye region detection and the SVM-based gaze recognition system. The experimental results are presented in Section 5, and we conclude in Section 6.

2 Related Work

2.1 Face and Eye Region Detection

The first step of gaze recognition is the detection of the face and eye regions in images. We use the AdaBoost algorithm based on Haar-like features for eye and face region detection. Figure 1 shows an example of the Haar-like features. Haar-like features are used as a kind of feature representation for object detection. The feature value is determined by subtracting the pixel sum of the black rectangular region from the pixel sum of the white region. The input image is scanned across location and scale [7].

Haar-like features are used to find the face and eye regions, and prototypes have been trained to accurately find the teeth region through the AdaBoost learning algorithm.

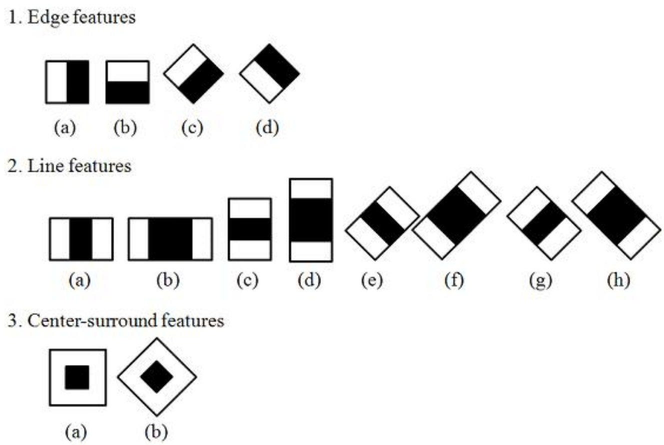


Fig. 1. Example of the Haar-like features

AdaBoost is a method to construct a strong classifier by combining weak multiple classifiers. A weak classifier is a single-layer perceptron, and is defined by Equation (1).

$$h_j = \begin{cases} 1 & if \quad f_j(x) < \theta_j \\ 0 & otherwise \end{cases}, \tag{1}$$

Where each weak classifier h_j is associated with a feature f_j and a threshold θ_j, and x indicates the size of the sub-window.

The Haar-like features obtained by the AdaBoost learning algorithm are classified in stages, as shown in Figure 2.

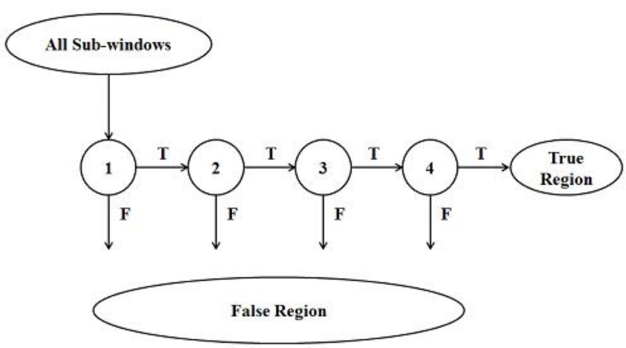


Fig. 2. Structure of cascade for region detection

2.2 PCA

Principal Component Analysis (PCA) is a well-known feature extraction and data representation technique widely used in the areas of pattern recognition, computer vision, and signal processing. The central underlying concept is to reduce the dimensionality of a data set whilst retaining the variations in the data set as much as possible [8]. Let an image that has $I(x, y)$ a two-dimensional M by L array of intensity values. In PCA, the corresponding x_i image is viewed as a vector with $M \times L$ coordinates that result from a concatenation of successive rows of the image. Denote the training set of N images by $X = \{x_1, x_2, \cdots, x_N\}$. Then, the corresponding covariance matrix is obtained by Equation (2).

$$C = \frac{1}{N}\sum_{N}^{1}(x_i \quad \bar{x})(x_i \quad \bar{x})^T \tag{2}$$

Where, $\bar{x} = 1/N\sum_{i=1}^{N} x_i$. Here, if $E = \{e_1, e_2, \cdots, e_D\}, D \le N$ are the eigenvectors of Covariance Matrix C that are sorted in descending order according to their corresponding eigenvalues, the feature vector $Y = \{y_1, y_2, \cdots, y_N\}$ is obtained by projecting images into the eigen spaces following the linear transformation.

$$y_j = E^T(x_i - \bar{x}) \tag{3}$$

The derived Y are gathered and stored in a database for use in the following recognition process. In the recognition process, each test image is normalized and represented as the new feature set. Then, a nearest neighbor classifier is usually used to measure the similarity between the training and test features. The nearest neighbor classifier is a simple classifier that requires no specific training phase. Here the distance between two arbitrary feature vectors, Y_i and Y_j is defined by Equation (4).

$$d(Y_j, Y_j) = \sqrt{\sum_{p=1}^{P}(Y_i^p - Y_j^p)^2} \tag{4}$$

Where, P denotes the vector dimension of the feature. Assume that the features of training images are $Y_1, Y_2, \cdots, Y_N$, where N is the total number of training images, and that each of these features is assigned a given identity (class) w_k . If a feature vector of a test image is given Y , the recognition result is decided finding the class, w_k that has a minimum distance as follows

$$d(Y, Y_r) = \min_j d(Y, Y_j), \quad and \quad Y_r \in w_k, \quad then \ Y \in w_k \tag{5}$$

2.3 DIE

In 1948, Shannon introduced a general uncertainty-measure on random variables that takes different probabilities among states into account [9]. Given events occurring with probability P, the Shannon entropy is defined as Equation (6).

$$H = \sum_{i=1}^{m} p_i \log\frac{1}{p_i} = \quad \sum_{i=1}^{m} p_i \log p_i \tag{6}$$

Shannon's entropy can also be computed for an image, where the probabilities of the gray level distributions are considered in the Shannon Entropy formula. A probability distribution of gray values can be estimated by counting the number of times each gray value occurs in the image and dividing those numbers by the total number of occurrences. An image consisting of a single intensity will have a low entropy value since it contains very little information. A high entropy value will be yielded by an image that has much different intensities. In this manner, Shannon's entropy is also a measure of dispersion of a probability distribution. A distribution with a single sharp peak corresponds to a low entropy value, whereas a dispersed distribution yields a high entropy value [10].

Shannon's entropy-based Difference Image Entropy (DIE) is computed with histogram levels of gray scaled-difference images which have peak positions from -255 to +255, to prevent of information sweeping. The average image from the M reference images is given in Equation (7).

$$S_{average} = \frac{1}{M}\sum_{m=1}^{M} S_m(x,y) \tag{7}$$

Where, parameter I_{input} is the input teeth image, and reference images m^{th} are represented with $S_m(x,y)$. The difference image D_{diff} is defined as Equation (8).

$$D_{diff} = I_{input} - S_{average} \tag{8}$$

Where, D_{diff} are computed using pixel subtraction between average image and input image. The DIE is defined as Equation (9).

$$E_g = \sum_{k=-255}^{255} P_k \log_2 \frac{1}{P_k} = -\sum_{k=-255}^{255} P_k \log_2 P_k \tag{9}$$

Where, E_g indicate DIE in gray images and p_k means probabilities of the frequency of the histogram in difference images. A probability p_k is defined as Equation (10).

$$P_k = \frac{a_k}{G_{(T)}} \tag{10}$$

Where, a_k indicates the frequency of the histogram from the -255 level to the +255 level, and the sum and total of each histogram in the difference image $G_{(T)}$ is given in Equation (11).

$$G_{(T)} = \sum_{k=-255}^{255} a_k \tag{11}$$

3 Proposed System

In this section, we describe the proposed SVM-based gaze recognition system using advanced eye region detection. For advanced eye region detection, we use a Gabor filter and binarization. Also, we implement our gaze recognition system using the SVM classification algorithm. Figure 3 shows the whole architecture for the proposed gaze recognition system.

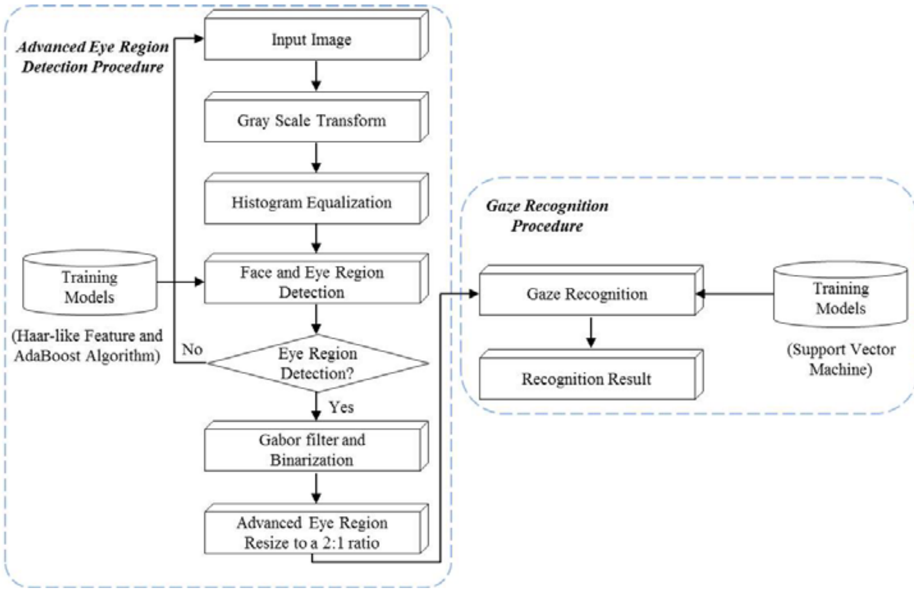


Fig. 3. Architecture for Proposed Gaze Recognition System

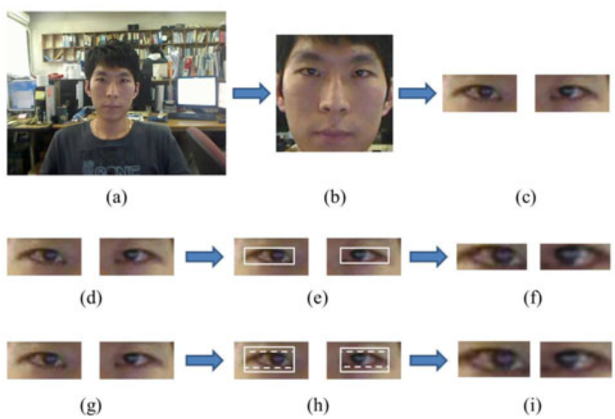


Fig. 4. Structure of cascade for region detection

3.1 Advanced Eye Region Detection

In this paper, we detected the eye region from the input image for gaze recognition. First, face and eye regions were detected from the input image using an AdaBoost algorithm based on Haar-like features. Then, we detected the corner of the eye and the iris using a Gabor filter and binarization. Finally, in the previous step the detected eye region image was doubled in size. Figure 3 shows the advanced eye region detection step. Figure 4(a) is the input image from the webcam, and Figures 4(b) and (c) are the detected face and eye region images using Haar-like features based the AdaBoost algorithm. Figure 4(d) is equal to the Figure 4(c), and the Figures 4(e) and (f) are the detected eye region image using Gabor filter and binarization. Figure 4(g) is equal to Figures 4(c) and (d), and Figures (h) and (i) are the detected eye region image using the advanced eye region detection method. In Figure (h), the dotted region equals to Figures (e) and (f).

3.1.1 Eye Corner Detection using Gabor filter

The Gabor filter works as a band pass filter for the local spatial frequency distribution, achieving an optimal resolution in both spatial and frequency domains [11]. The Gabor filter has been extensively used as a feature extractor for various applications [12], [13]. The Gabor filter is defined as Equations (12), (13), and (14).

$$h(x, y, \varphi, f) = \exp\{-\frac{1}{2}[\frac{x_\varphi^2}{\sigma_x^2} + \frac{y_\varphi^2}{\sigma_y^2}]\}\cos(2\pi f x_\varphi) \tag{12}$$

$$x_\varphi = x\cos\varphi + y\sin\varphi \tag{13}$$

$$y_\varphi = -x\cos\varphi + y\sin\varphi \tag{14}$$

Where, φ is the orientation of the Gabor filter and f is the frequency of a sinusoidal plane wave. Also, σ_x and σ_y are the sigma of the Gabor envelope which is set to a constant value.

In this paper, we detected the eye region image by dividing it into two parts, and applying the Gabor filter for eye corner detection.

Figure 5 shows the eye corner detection step using the Gabor filter. Figure 5(a) is the detected eye region image using Haar-like features based on the AdaBoost algorithm, and (b) is the divided eye region image. Figure 5(c) is the input image, and Figure 5(d) is the Gabor kernel. Figures 5 (e) and (f) are the response and magnitude images, and Figure 5 (g) is the binary image of Figure 5(f). Figure 5(h) shows the detected eye corner.

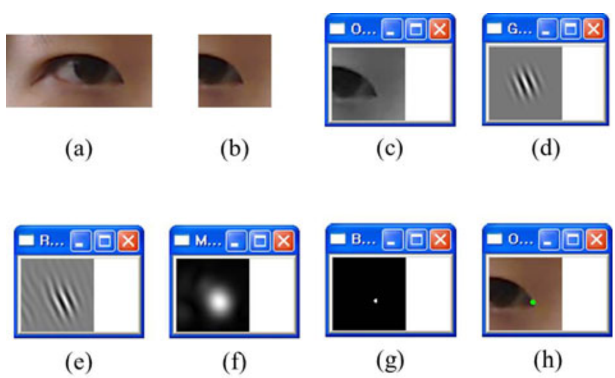


Fig. 5. Eye corner detection step using Gabor filter

3.1.2 Iris Detection Using Binarization

In this paper, we detect the iris from the detected eye region image using binarization. Figure 6 shows the iris detected method and examples. The detected eye region image is converted to the threshold image. The threshold conversion is performed according to Equation (15).

$$g(x,y)=\begin{cases} 0 & if \quad f(x,y)\le T \\ 255 & if \quad f(x,y)>T \end{cases} \tag{15}$$

Where, $f(x,y)$ and $g(x,y)$ are an intensity value of the detected eye region image and the threshold image at position (x,y), respectively, and T is the threshold value [14]. The threshold image is divided into several small blobs. Consequently, these blobs are mixed for iris detection as shown in Figure 6(a).

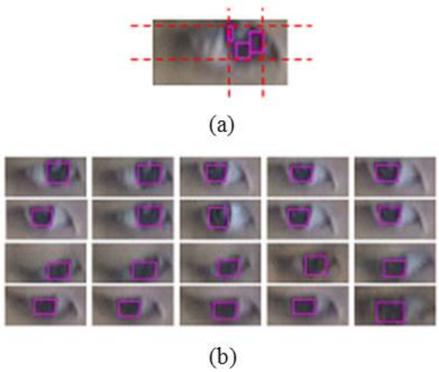


Fig. 6. Iris detection method and examples

3.2 SVM-Based Gaze Recognition System

In this paper, we propose and implement a SVM-based gaze recognition system. Since the SVM was proposed by Cortes and Vapnik [15], it has been successfully applied in many pattern recognition problems. SVM is a linear classifier in high dimension space with minimal structural risk. Suppose the pattern is given as (x,y), where x is the feature vector and $y \in -1,1$ is the class label. The linear decision function in SVM is defined as Equation (16).

$$f(x)=sign(w \bullet x+b) \tag{16}$$

Where, weight w is the linear combination of Support Vectors (SVs).

$$w=\sum_{i=1}^{l} a_i x_i \tag{17}$$

The decision boundary is determined by maximizing the structural margin between the two classes. The optimization problem is equivalent to the quadratic programming problems [13]. The kernel function in SVM plays a role if mapping a feature vector to a higher dimensional space and dot product. By replacing the dot products with the kernel function $k(x, x_i)$, we obtain the linear discriminant function in high dimension feature space.

$$f(x) = sign(\sum_{i=1}^{l} a_i y_i k(x, y) + b) \tag{18}$$

The most often used kernel functions in eye recognition are polynomial with degree d, Gaussian RBF and Multi-Layer Perceptron [16]. A 2-d polynomial SVM is trained to recognize eyes in images. The eye candidates are cut from images and preprocessed, and they are then sent to the SVM. The false positives are fed back to the SVM and used as negative samples for later training. The single SVM shows good recognition rate for high resolution images. However, its performance deteriorates for mixed-quality images.

4 Experiments and Results

The proposed gaze recognition system was implemented using Microsoft Visual C++ 6.0 on a PC with a webcam. We used 14,400 training images and 3,600 testing images, a total of 18,000 eye region images in total. To evaluate the performance of proposed gaze recognition system, we implemented a second gaze recognition system based on DIE and PCA in order to compare it with our proposed system. Also, we used the same eye region images for three different eye region detection methods. The first method detects eye regions using Haar-like feature-based AdaBoost algorithm. The second method detects a Gabor filter and binarization-based eye region using detected eye region image in first method. The third method used a resized image with a 2:1 ratio of detected eye region images in second method. In third method, the reason why we resized the eye region image which was detected through second method to a 2:1 ratio is that there was a distortion in the eye area in the original image when we normalized the image for gaze recognition.

In the case of the DIE-based gaze recognition system, the recognition rate for looking at the four large areas was 75.36%, and the recognition rate for looking at the nine smaller areas was 40.17% using detected eye region image through method 1. When we detected eye region images through method 2, the rates were 74.94%, and 38.44% for the same tests. When using detected eye region images through method 3, our proposed advanced eye region detection method, the recognition rates were 77.25% and 41.89% for the same tests. Table 1 shows the experimental results for DIE-based gaze recognition.

For the PCA-based gaze recognition system, the recognition rate for the four large areas was 81.53%, and for the nine smaller areas was 53.11% using detected eye region image through method 1. When we used detected eye region image through method 2, recognition rates were 81.17% and 52.75% for the same tests. When using detected eye region image through method 3, our proposed advanced eye region detection method, the recognition rates were 83.92% and 54.94% for the same tests. Table 2 shows experimental result for PCA-based gaze recognition.

The experimental result of the performance evaluation using the proposed SVM-based gaze recognition system is shown in Table 3. When using the detected eye region image through method 1, recognition rates were 94.42% and 81.33%, and when using the detected eye region image through method 2, recognition rates were 97.33% and 92.58%. When using the detected eye region image through method 3, we proposed an advanced eye region detection method, and recognition rates were 97.81% and 92.97%.

Table 1. Experimental results for DIE-based gaze recognition system

	Method1	Method2	Method3
4 Directions	75.36%	74.94%	77.25%
9 Directions	40.17%	38.44%	41.89%

Table 2. Experimental results for PCA-based gaze recognition system

	Method1	Method2	Method3
4 Directions	81.53%	81.17%	83.92%
9 Directions	53.11%	52.75%	54.94%

Table 3. Experimental results for proposed SVM-based gaze recognition system

	Method1	Method2	Method3
4 Directions	94.42%	97.33%	97.81%
9 Directions	81.33%	92.58%	92.97%

5 Conclusions

In this paper, we propose and implement a SVM-based gaze recognition system using an advanced eye region detection method and a webcam. Accurate eye region detection is essential for high performance of gaze recognition. So, we detect eye corners and irises using a Gabor filter and binarization for accurate eye region detection. We implement a PCA and DIE-based gaze recognition system in order to evaluate our proposed system. And we performed the same experiments for all three eye region detection methods using the same images. The proposed system divided the PC screen into four large squares. A second method divided the PC screen into nine smaller squares for gaze recognition. We used 14,400 training images and 3,600 testing images, for a total of 18,000 eye region images. In the experimental results, the proposed gaze recognition system using a Gabor filter and binarization-based advanced eye region detection method which showed the highest performance.

From the experimental results, we can confirm that the recognition rate of the proposed SVM-based gaze recognition system was better than general approaches such as the PCA and DIE-based gaze recognition systems. In addition, we validated the effectiveness of the proposed advanced eye region detection method through experimental results.

For future work, we expect to apply a multi-modal human interface using the senses such as speech and gestures based on the proposed gaze recognition system. Also, we plan in future work, to enhance the recognition rate and more direction of recognition through improved algorithms or optimization processes. Furthermore, we will investigate head movement correction and 3-D gaze recognition.

Acknowledgments. This research was supported by MKE, Korea under ITRC NIPA-2010-(C1090-1021-0008), and Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2010-0021411).

References

1. Djamasbi, S., Siegel, M., Tullis, T., Dai, R.: Efficiency, Trust, and Visual Appeal: Usability Testing through Eye Tracking. In: Proceedings of the 2010 43rd Hawaii International Conference on System Sciences, pp. 1–10 (2010)
2. Handa, S., Ebisawa, Y.: Development of head-mounted display with eye-gaze detection function for the severely disabled. In: Proceedings of the IEEE Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems, pp. 140–144 (2008)
3. Karpov, A., Ronzhin, A., Nechaev, A., Chernakova, S.: Multimodal System for Hands Free PC Control. In: EUSIPCO 2005 (2005)
4. Kinoshita, K., Ma, Y., Lao, S., Kawade, M.: A Fast and Robust 3D Head Pose and Gaze Estimation System. In: Proceedings of the 8th International Conference on Multimodal Inter-Faces, pp. 137–138 (2006)
5. Hutchinson, T.E., White, K., Worthy, J.R., Frey, A.: Human Computer Interaction using Eye-Gaze Input. IEEE Transaction on Systems, Man, and Cybernetics 19, 1527–1533 (1989)
6. Kocejko, T., Bujnowski, A., Wtorek, J.: Eye Mouse for Disabled. In: Human System Interactions 2008, pp. 199–202 (2008)
7. Viola, P., Jones, M.J.: Robust real-time object detection. Technical Report Series. Compaq Cambridge research Laboratory (2001)
8. Turk, M., Pentland, A.: Eigen faces for recognition. Journal of Cognitive Neuro-Science, 71–86 (1991)
9. Shannon, C.E.: A Mathematical Theory of Communication. The Bell Systems Technical Journal 27, 379–423 (1948)
10. Loutas, E., Pitas, I., Nikou, C.: Probabilistic multiple face detection and tracking using entro py measures. IEEE Transaction on Circuits and Systems for Video Technology 14(1), 128–135 (2004)
11. Daugman, J.G.: Uncertainty Relations for Resolution in Space, Spatial Frequency, and Orientation Optimized by Two-Dimensional Visual Cortical Filters. Journal of the Optical Society of America 2, 1160–1169 (1985)
12. Huang, L.L., Shimizu, A., Kobatake, H.: Robust Face Detection using Gabor Filter Features. Pattern Recognition Letters 26, 1641–1649 (2005)

13. Bhuiyan, A., Liu, C.H.: On Face Recognition using Gabor Filters. In: Proceeding of World Academy of Science, Engineering and Technology, pp. 51–56 (2007)
14. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Addison-Wesley, Reading (1992)
15. Cortes, C., Vapnik, V.: Support-vector network. Machine Learning 20, 273–297 (1995)
16. Osuna, E., Freund, R., Girosi, F.: Training support vector machines: an application to face detection. In: Proceeding of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 130–136 (1997)

Anti Jamming – Based Medium Access Control Using Adaptive Rapid Channel Hopping in 802.11: AJ-MAC

Jaemin Jeung, Seungmyeong Jeong, and Jaesung Lim*

Graduate School of Ajou University, South Korea
{mnsg,aflight,jaslim}@ajou.ac.kr

Abstract. 802.11 networks have inherent security weaknesses due to wireless characteristics, especially physical layer jamming. Recent research shows channel hopping is a better way to combat jamming or interference compared to changing 802.11 operational parameters (such as, clear channel assessment threshold, rate, and packet size). However frequent channel hopping decreases network throughput, and intermittent channel hopping raises the jammer's detection probability. We introduce an Adaptive Rapid Channel Hopping method using Dwell Window (DW) to mitigate these defects. This is a novel concept to adjust transmission time based on the jammer's ability. That is, if a jammer successfully attacks a channel, the channel's DW is decreased to reduce the jammer's detection probability. In the case where there is no jamming, DW is increased to raise network throughput. In addition to this scheme, we introduce a Deception Mechanism that is another novel concept to make a jammer attack an unnecessary channel for a high throughput and a low probability of detection. Numerical analysis and simulation results show that the proposed scheme is more effective than those of prior studies.

Keywords: Anti-jamming, channel hopping, Medium Access Control.

1 Introduction

802.11 networks have been widely used in many areas due to their mobility and convenience. Furthermore, a drastic increase of smart phones using WiFi increases the prevalence of 802.11 networks. However, as 802.11 networks become more prevalent, and many areas become more dependent on the networks. Many problems could be caused by security weaknesses, especially physical layer jamming. Because 802.11 networks use Industrial Scientific Medical (ISM) bands that an adversary can easily detect and jam using the same protocol. Therefore, much malicious code can be developed by modifying the 802.11 rules, such as Distributed Coordination Function (DCF), to attack this vulnerability. The modified codes can continuously transmit a large number of frames, regardless of these rules. This decreases network throughput.

* This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2011-C1090-1121-0011).

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 70–82, 2011.

Unfortunately, 802.11 networks were not designed to combat physical layer jamming, so the networks cannot mitigate this jamming. Therefore, mitigating physical layer jamming is very important to securely use the convenient 802.11 networks.

Research to mitigate physical layer jamming and interference [1-4] concluded that channel hopping is a better way to combat jamming or interference compared to changing 802.11 operational parameters, such as clear channel assessment (CCA) threshold, rate, and packet size. There are two channel hopping methods. The first approach, proactive rapid channel hopping, performs rapid channel hopping, regardless of channel status [1-2]. The second approach, reactive channel hopping hops to good channels, only after identifying whether the current channel is good [3-4]. The first approach decreases network throughput, because sometimes it is likely to hop more often than necessary, as well as it unnecessarily hops without a jammer. Conversely, the second approach can mitigate these defects. However, it cannot prevent a smart jammer who jams the detected channel after sensing. As the second approach, removing the jammed channels or bad channels from the hopping channel set cannot find a good channel under the smart jammer. Briefly, the earlier studies had some defects. That is, proactive rapid channel hopping methods decrease network throughput, while reactive channel hopping methods increase the jammer's detection probability.

In this paper we introduce an Adaptive Rapid Channel Hopping method using Dwell Window (DW) to mitigate these defects. This is a novel concept to adjust each channel's transmission time based on the jammer's ability. That is, if the channel state becomes bad due to jamming or interference, the channel's DW is decreased, and vice versa. However, this method will likely decrease network throughput under the smart jammer, because the network cannot access the channel during the remaining DW after the jammer successfully jams the channel. We introduce a Deception Mechanism, another novel concept to make the jammer continuously attack an unnecessary channel, to mitigate this throughput degradation. Thus, this increases network throughput and decreases the jammer's detection probability. The process is as follows. Assuming that all devices including the jammer are Time Division Duplex (TDD) systems using half duplex; only one device can transmit frames at a time. If the jammer attacks the detected channel after sensing a transmitting client (tx-client), receiving clients (rx-client) and AP hop to a predetermined next channel using a notification method, such as a jam announcement signal. Meanwhile, the tx-client and the jammer cannot hear the jam announcement signal due to their TDD rules. Furthermore, the tx-client cannot know if the jammer is operating during its transmission time for the same reason. The jammer must sense again the jammed channel after jamming, assuming that the jammer periodically stops jamming and senses again the channel to check if the network hops to another channel. At this time, the tx-client determines that the channel is available and retransmits frames based on 802.11 DCF rules; this deceives the jammer. That is, the jammer thinks that the network does not hop to another channel; this makes the jammer again attack the channel. Meanwhile, the tx-client hops to the predetermined next channel after the DW expires. Thus, the jammer unnecessarily attacks the channel that legitimate devices are not using.

The proposed scheme using DW and Deception Mechanism can increase network throughput and decrease the jammer's detection probability under the smart jammer, as well as any interference. These characteristics lead us to name our proposed scheme Anti Jamming – based Medium Access Control (AJ-MAC). Numerical analysis and simulation results show AJ-MAC is more effective than the approaches of prior research.

The remainder of this paper is organized as follows. Section 2 describes the background of the proposed scheme and related work about proactive and reactive channel hopping methods. We introduce the jammer model and AJ-MAC using DW and Deception Mechanism in Section 3. Numerical analysis and simulation results are presented in Section 4. Finally we conclude this paper and outline directions for future work in Section 5.

2 Background and Related Work

Channel hopping is a little bit different from frequency hopping and channel change. Frequency hopping depends on rapidly changing the transmission frequency that can be no wider than 1 MHz by the Federal Communications Commission (FCC) regulations, while channel hopping depends on changing the transmission channel used on commodity Network Interface Cards (NICs). Conversely, a channel change typically occurs only in response to failures in current 802.11 systems and hops to a good channel, while channel hopping hops to several channels in a hopping set at a moderate rate. Briefly, channel hopping lies between frequency hopping and channel change. Considering prior research, these channel hopping methods are generally categorized by proactive rapid channel hopping and reactive channel hopping. The first approach performs rapid channel hopping, regardless of channel status. The second approach hops to good channels, only after identifying if the current channel is good. We consider each in turn below.

2.1 Proactive Rapid Channel Hopping

The authors in [1] proposed a rapid channel hopping scheme to combat interference (e.g., cordless phones, Bluetooth headsets, Zigbee, camera jammer) in the 802.11 overlapping channels. They showed that channel hopping is a better way to combat interference than changing 802.11 operational parameters (e.g., clear channel assessment threshold, rate, and packet size). They also showed that three interferers on all orthogonal channels do not shut down a channel hopping link in 802.11 b. However, the scheme hops more often than necessary, increasing channel switching time. That degrades the network throughput. Conversely, the authors in [2] proposed a rapid channel hopping scheme to combat a smart jammer that attacks the detected channel after repeatedly sensing if the channel is used. They defined the smart jammer and found the optimal transmission time based on the jammer. However, it cannot adjust the transmission time based on the jammer's ability but just uses the fixed transmission time. Furthermore, if the channel is attacked by the smart jammer, the channel cannot be totally used during this time. Thus, this scheme also degrades network throughput. Besides [1-2], authors in [3-4] also introduced hopping methods to improve the capacity or fairness. However, their schemes have some defects due to network throughput degradation or high detection probability.

2.2 Reactive Channel Hopping

802.11h is one of the reactive channel hopping methods. That is, the network can hop to new channel using Dynamic Frequency Selection (DFS) algorithm, after channel quality degradation occurs. DFS has four different states: normal operation, channel DFS test, full DFS test, and frequency change [3]. This mechanism is useful for automatic frequency planning under interference. However, it is very weak under the smart jammer, since four states of the DFS are likely to be infinitely repeated. Thus, the network cannot find a good channel. On the other hand, authors in [4] showed four different adaptive frequency hopping methods based on 802.15: Adaptive Frequency Hopping (AFH), Adaptive Frequency Rolling (AFR), Adaptive Frequency Rolling with Probing (AFR-P), and Dynamic Adaptive Frequency Hopping (DAFH). Although frequency hopping is a little different from channel hopping, the strategies to combat interference can apply to reactive channel hopping methods. First, AFH removes bad channels from the hopping set, after monitoring channel quality. The strategy of "bad channel removal" is not good under the smart jammer, as the smart jammer makes the available hopping channels become smaller. Thus, the jammer can easily detect the network and degrade network throughput. The strategy of "bad channel removal" could be more severe, considering that the 802.11b has only three orthogonal channels and the 802.11a has twelve orthogonal channels. Second, AFR changes the hopping set randomly, as soon as channel quality degradation occurs. Each network has a randomized channel quality threshold to avoid the simultaneous change of different wireless networks. Third, AFR-P is an extension of AFR. This scheme detects which channels are bad, and then removes the detected bad channel, and performs AFR with the reduced hopping set. However, these two schemes are not good under the smart jammer, since the smart jammer makes the networks frequently change the hopping set. Thus, network throughput drastically decreases. Finally, DAFH is a randomized binary splitting of the hopping set under interference or jamming. If the interference exceeds the threshold, the network is triggered and splits its hoping set with the binary method. Then, it randomly selects the new hopping set to be either the left or right half of the current hopping set. After time *T*, the network doubles its hopping set to recover its original hopping set. This method is good in achieving fairness. However, the smart jammer makes the hopping set very small. This increases the jammer's detection probability.

Briefly, the prior research has defects. That is, the proactive rapid channel hopping methods decrease the network throughput, while the reactive channel hopping methods increase the jammer's detection probability. Therefore, increasing the network throughput and decreasing the detection probability are crucial to securely and conveniently use 802.11 networks under the jammer or interference.

3 Anti Jamming-Based Medium Access Control (AJ-MAC)

3.1 Jammer Model

Generally, a jamming attack is achieved by introducing a source of noise strong enough to significantly reduce the capacity of the channel. A jammer with unlimited resources can always successfully attack any wireless communication by flooding the entire spectrum. In that case, it is obviously impossible to mitigate the jammer's attack. However, it is hard for a jammer to have unlimited resources in the real world due to broadband characteristics, powers, costs, and so on. Thus, it is beneficial to make a plan for increasing resilience. It is reasonable to be a resource-constrained jammer; especially an 802.11 based jammer may be restricted to a configuration similar to that of a legitimate 802.11 based station. It is more reasonable to be an 802.11 based jammer who modifies the software like a virus, jams the wireless networks, regardless of 802.11 rules. Furthermore, it is hard to detect this jammer, since it may disguise itself with an 802.11 based hardware, such as a smartphone, a laptop, or a notebook. These reasonable jammer models make mounting a defense possible.

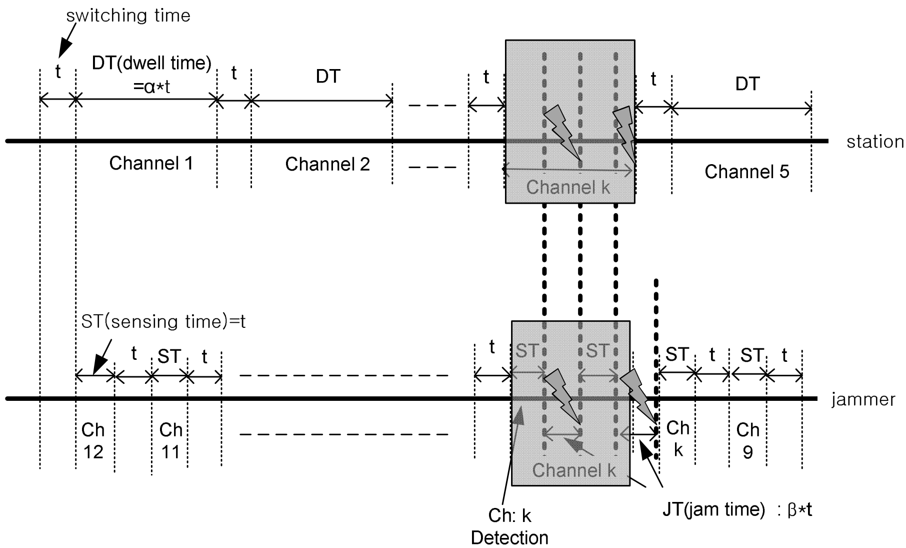


Fig. 1. Smart jammer model

In this paper, our jammer model is a smart jammer who transmits back to back packets on the detected channel after scanning all of the channels, until the legitimate communication hops to a new channel. Of course the smart jammer and stations are resource-constrained devices due to using an 802.11 based hardware. We assume that the stations can hop between channels by hopping sequence, and the jammer cannot estimate the sequence due to pseudo-random hopping characteristics. Fig. 1 shows the smart jammer model. The jammer quickly scans the entire spectrum of channels by its own channel scanning sequence, since it does not know the channel hopping sequence of the legitimate communications. The jammer takes a switching time and sensing time to verify one channel. Channel changing in a conventional 802.11 hardware takes some time. This is known as the switching time. Channel sensing to find a legitimate communication takes some time; this is the sensing time. In [2], it was revealed that these take a few milliseconds. Therefore, we let the switching time and sensing time be same t so as to simplify our jammer model. The station that uses a channel hopping method transmits data during a dwell time (DT=$\alpha \times t$) after switching time t.

In the gray rectangle in Fig.1, if the jammer detects a channel k, it jams the channel during a jam time (JT=$\beta\times t$). After jamming during JT, it checks if the legitimate communication hops to a new channel during ST. If the communication is still there, the jammer attacks the channel again during JT. Otherwise it hops to a new channel using its scanning channel sequence to find the legitimate communication.

3.2 AJ-MAC Scheme

We propose Anti-Jamming based Medium Access Control (AJ-MAC) that can mitigate physical layer jamming using an adaptive rapid channel hopping method. We introduce Dwell Window (DW). This is a novel concept to adjust each channel's transmission time based on the jammer's ability to achieve an adaptive anti-jamming property. Fig. 2 depicts AJ-MAC DW that has some time slots from t to αt. The time slots α is related to the network throughput and the jammer's detection probability. If the time slot α is too small, then frequent channel changing will reduce network throughput. If timeslot α is too large, then the smart jammer will easily find and successfully jam the network. Therefore, timeslot α should be determined by the jammer's ability. That is, if the jammer's ability to find and jam is high**,** the timeslot α is set to be smaller. Otherwise, the timeslots α is set to be larger to increase network throughput. After transmitting by channel-A during DW, the AJ-MAC switches to channel-B during a switch time t. As soon as the AJ-MAC hops to a new channel-B, it transmits a beacon to broadcast some hopping parameters, as well as to synchronize all legitimate devices.

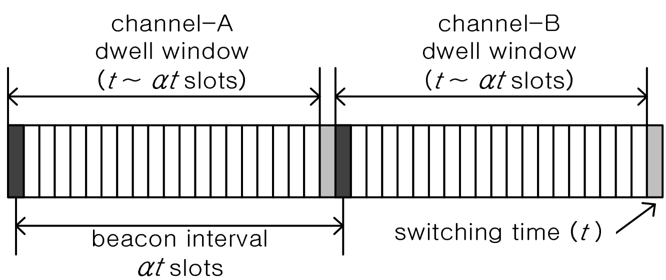


Fig. 2. AJ-MAC Dwell Window

After setting timeslots α, AJ-MAC adjusts DW within the time based on the attacks. Adjusting DW is more effective than adjusting all channels' α every hop time. Because the latter should change all channels' α, even though the jammer attacks successfully just a few channels; this increases the switching times and degrades throughput. Fig. 3 shows AJ-MAC increases and decreases the jammed channel's DW to mitigate the jammer's detection probability. The basic strategy applied in AJ-MAC is a binary reduction and doubling. In Fig. 3, if the jammer detects and jams channel-A, the AJ-MAC splits the DW in half, and then selects a good packet error rate (PER) part for the next hopping sequence's channel-A. On hopping to channel-A again, the AP informs the selected part (white rectangle in Fig. 3) to the stations. If the jammer attacks again the selected part of channel-A, AJ-MAC splits the selected part further.

If the DW to split becomes an odd number, AJ-MAC selects a part whose slots are odd. This process will be continued until the DW becomes timeslot *t*. A doubling mechanism is complementary to reduction. If the jamming is not detected during the sensing time (dark gray rectangle Fig. 3), the AJ-MAC doubles its reduced DW for the next hopping sequence's channel-A. On hopping to channel-A again, AP notifies the doubling information to the stations. This process will be continued until the DW becomes timeslot α. This reduction and doubling mechanism also apply to timeslot α. That is, if there are no jamming attacks during a threshold, AJ-MAC doubles timeslot α to increase network throughput by reducing channel switching time. If all channels' DW size becomes a threshold, all channels' timeslot α is reduced by half to mitigate the jammer's detection probability.

AJ-MAC can overcome some defects of proactive rapid channel hopping and reactive channel hopping using the reduction and doubling mechanism of the DW. That is, the needless hopping of the former and bad channel removal problems of the latter are reduced by the adaptive rapid channel hopping method using the reduction and doubling mechanism.

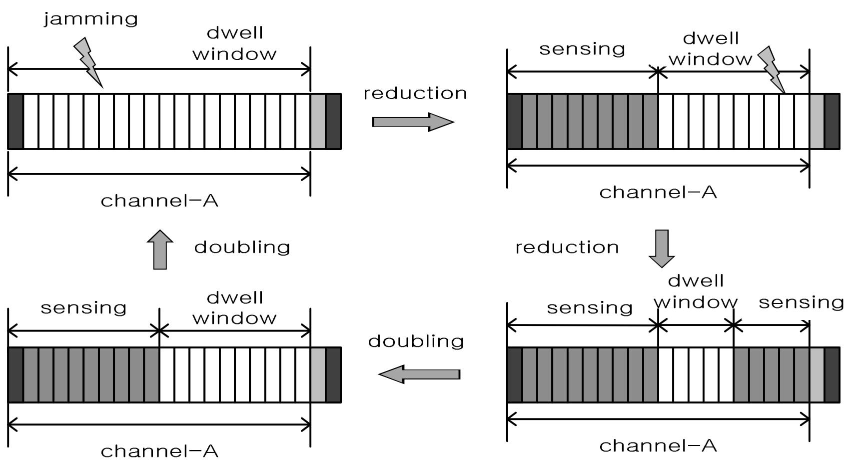


Fig. 3. AJ-MAC Dwell Window reduction and doubling mechanism

3.3 Deception Mechanism

The adaptive rapid channel hopping mechanism of AJ-MAC can increase network throughput and decrease the jammer's detection probability. However, there is one defect when the jammer detects and jams the channel. That is, once the jammer jams the detected channel, all legitimate devices including AP and stations cannot communicate to each other during the channel's DW, decreasing network throughput. Therefore, we introduce a Deception Mechanism to mitigate this throughput degradation. This is another novel concept to make the jammer continuously attack an unnecessary channel during the DW.

Fig. 4 and Fig. 5 show the AJ-MAC Deception Mechanism and its network topology, respectively. There are 5 entities in fig. 5: an AP, the jammer, an rx-client who is receiving frames, a tx-client who is transmitting frames, and a hidden terminal who cannot hear the jam signal. Assuming that all entities including the jammer are Time Division Duplex (TDD) systems using half duplex, only one device can transmit frames at a time. Therefore, the tx-client cannot know if the jammer is attacking the transmitting frames. Of course the jammer also cannot hear if the AP sends a jam announcement signal on transmitting jam frames due to the same reason; this makes mounting the Deception Mechanism possible. The process is as follows. If the jammer detects and jams the channel on which the tx-client is transmitting, the rx-client, hidden terminal and AP hop to a predetermined next channel using the notification method, such as a jam announcement signal with a high power and good channel coding scheme. Thus, all entities except for the tx-client and jammer can communicate by the hopped next channel (channel-B in Fig. 4) during the jam time (a checked rectangle in Fig. 4). Meanwhile, the tx-client and jammer cannot hear the announcement signal, since they are all transmitting by the half-duplex rule, as in our smart jammer model, the jammer periodically must stop jamming and sense the channel again, because it has to check if the network hops to another channel. At this time (white rectangle between jam time in Fig. 4), the tx-client determines that the channel is available, and then retransmits the frames not receiving an ACK from the AP according to 802.11 DCF rules; thus, deceiving the jammer. That is, the jammer thinks that the network does not hop to another channel, since the jammer could detect the legitimate communication on that channel; this makes the jammer attack again the channel. Meanwhile, the tx-client hops to the predetermined channel-B after the DW expires. Therefore, the jammer unnecessarily attacks the channel that any legitimate device does not use. In case the AP is jammed, the Deception Mechanism is partially applied to this case. If the AP finishes the transmission and starts receiving during the jam time, the AP can detect the jammer easily by checking the Access Control List (ACL), duration field, sequence number, and so on [5]. If the jamming is detected, the AP sends a jam announcement signal; this makes all entities, except for the jammer, hop to the next channel-B. Conversely, if the AP cannot finish a transmission during the jam time, it cannot send a jam announcement signal; this prevents all entities from hopping to a new channel until the DW expires. This degrades network throughput. However, it is likely to be impossible for the AP to transmit continuously during the jam time due to fragmentation in 802.11 MAC [6].

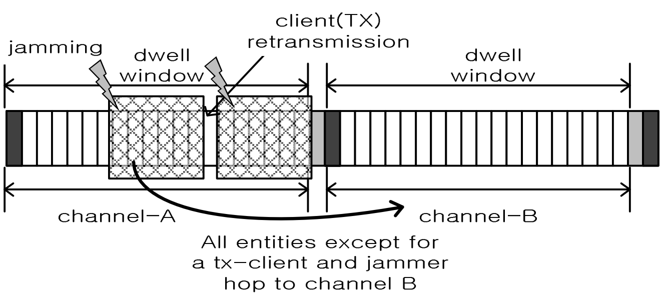


Fig. 4. AJ-MAC Deception Mechanism

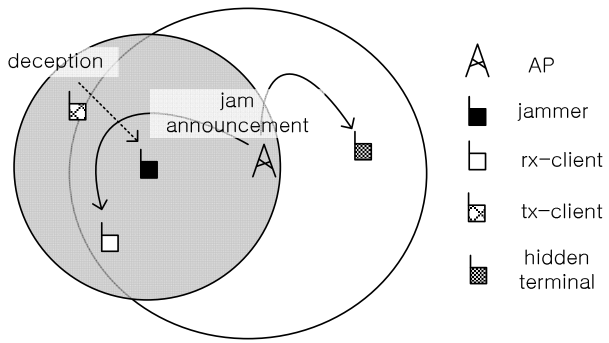


Fig. 5. Network topology for AJ-MAC Deception Mechanism

4 Numerical Analysis and Simulation Results

4.1 Numerical Analysis

In this section, we provide numerical analysis for a throughput comparison of a channel hopping method and the proposed AJ-MAC scheme under a smart jammer given the DW size. We derive formulas that show some characteristics about a channel hopping method and the AJ-MAC scheme. We reasonably propose assumptions and modify the jammer model in section 3.1 to simplify the numerical analysis. First, the jammer transmits back to back packets on the detected channel after sensing the channel, until the DW is expired. That is, the client cannot successfully transmit any frame after being detected. Second, the jammer and client are independent. Therefore, the client cannot know the jammer's scanning pattern and the jammer also cannot know the client's hopping sequences. Third, the jammer perfectly knows when the DW starts. Finally, we use a normalized throughput rate that is without channel hopping. Therefore the normalized throughput is the fractional rate with respect to no channel hopping. These assumptions make the numerical analysis possible by a probability problem.

As in Fig 1 in section 3.1, time t is the required time for the jammer and client to switch to a different channel. Therefore the jammer cannot jam any channel, and the client cannot transmit any frame during time t. Supposing the client hops to another channel after transmitting on the current channel during DW ($\alpha \times t$), we can calculate the normalized throughput rate like $f(\alpha) = \alpha t/(\alpha+1)t$ without a jammer. A white circle in Fig 6 shows the normalized throughput curve. If the client hops every beacon interval (100 ms, time t = 5 ms), the throughput can be up to 95% at DW 20 (100 ms = $\alpha \times t$ = 20×5 ms).

In the channel hopping model under the smart jammer, we have to analyze the characteristics of the jammer's behavior. The jammer can detect N number of channels by a pseudorandom selection during DW ($\alpha \times t$). N can be computed as

$$N = \lfloor \alpha t/(ST+1)t \rfloor = \lfloor \alpha/(ST+1) \rfloor \qquad (where \quad 0 \le N \le L) \qquad (1)$$

In (1), time αt is DW size, time ST is the jammer's sensing time, and L is the total number of channels. The probability that the jammer can detect the client's channel after sensing an n_{th} channel is expressed as

$$P_1 = \frac{1}{L},\ P_2 = \frac{L-1}{L}\times\frac{1}{L-1},\ P_3 = \frac{L-1}{L}\times\frac{L-2}{L-1}\times\frac{1}{L-2}$$

$$P_n = \frac{L-1}{L}\times\frac{L-2}{L-1}\times\ldots\times\frac{L-n+1}{L-n+2}\times\frac{1}{L-n+1} = \frac{1}{L} \quad (where\quad n = 1,2,3,\ldots,N) \tag{2}$$

The client's jammed time after being detected in an n_{th} channel is expressed as

$$T_n = \alpha t - nST - (n-1)t \quad (where\quad n = 1,2,3,\ldots,N) \tag{3}$$

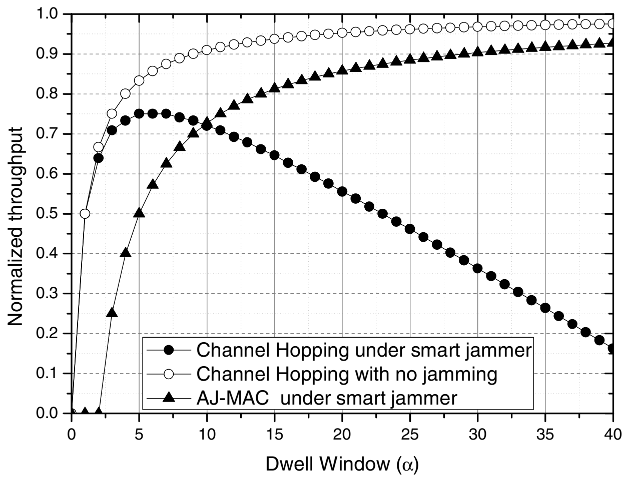


Fig. 6. Normalized throughput versus Dwell Window size

The client's average jammed time during DW size is expressed as

$$E(t) = \sum_{n=1}^{N} T_n \times P_n \quad (where\quad n = 1,2,3,\ldots,N) \tag{4}$$

Finally, the normalized throughput of the channel hopping method under the smart jammer can be obtained as

$$Thrpt_{channel\ hopping} = \frac{\alpha t - E(t)}{(1+\alpha)t} \tag{5}$$

A black circle in Fig. 6 shows the throughput plot under the smart jammer given $ST = t$, L=12 (the number of channels in 802.11a; in the case of 802.11b, the orthogonal channel is only 3, which increase the jammer's detection probability. Therefore we consider 802.11a devices). The curve shows that the throughput achieved is up to 75% by the channel hopping method with the right DW size under the smart jammer. However, the method cannot increase the throughput more than 75%, since the throughput is drastically lowered as DW size increases. Conversely, our proposed AJ-MAC can increase the throughput more than 75% under the smart jammer. The throughput of the AJ-MAC can be obtained as

$$Thrpt_{AJ-MAC} = \frac{\alpha t - \gamma t}{(1+\alpha)t} \tag{6}$$

The time γt is a summation of jamming detection, announcement, and switching time. As in section 3.3, Deception Mechanism, AJ-MAC can increase the DW size to achieve a good throughput, since it can deceive the smart jammer. The characteristics can make the AJ-MAC transmit frames during the time $(\alpha t - \gamma t)$. The announcement and switching time are fixed, so the time γt is relative to jamming detection time. We can easily detect physical layer jamming using a combination of signal strength variation and a checksum of PLCP (Physical Layer Convergence Protocol) header in [7], [8]. A black triangle in Fig. 6 shows AJ-MAC's throughput plot under the smart jammer given $ST = t$, L=12, and $\gamma t = 2t$. Supposing that the client hops every beacon interval (100ms, time t = 5ms), the throughput can be up to 85%.

4.2 Simulation Results

In this section, we provide the performance analysis to compare throughput of channel hopping methods and the proposed AJ-MAC scheme using the OPNET modeler 14.5. In the simulation environment, we do not consider any layer above the MAC to observe clearly the MAC layer behavior under the smart jammer. We modify the Frequency-Swept Jammer model in OPNET modeler to realize the smart jammer of section 3.1. In the case of proactive channel hopping, we make a station randomly hop to another channel after a fixed time. The Dynamic Adaptive Frequency Hopping (DAFH) strategy of section 2.2 applies to this scheme for reactive channel hopping, since the DAFH scheme can inherently combat interference [4]. Finally, we let a station have the novel concepts of the DW and Deception Mechanism for the AJ-MAC. Other simulation assumptions and conditions are the same as in section 4.1.

Table 1. Simulation Parameters

Parameters	Value
Jammer's active time	2nd~10th second
Jam Time (JT)	50 msec
Sensing Time (ST)	5 msec
Switching time	5 msec
Dwell Window (DW) size	100 msec
Beacon interval	100 msec
Number of orthogonal channels	12

Table 1 shows the simulation parameters. The smart jammer is active between the 2nd and 10th second, jam time is 50 ms, sensing and channel switching time are 5 ms, DW size & Beacon interval are 100 ms, and the number of orthogonal channels is 12. Fig. 7 shows the simulation topology. The network throughput is calculated at the AP. Other conditions are the same as in section 3.3. In Fig. 8, the curves show the normalized throughput for different channel hopping schemes. The throughput of the proactive channel hopping (black circle in Fig. 8) is decreased to 67%, as the jammer becomes active. In case of the reactive channel hopping, this is more serious than in the proactive scheme. This scheme makes the available hopping channels small, which increases the probability of the jammer's detection. The throughput is drastically reduced to 34% (white triangle in Fig. 8). However, AJ-MAC (black triangle in Fig. 8) can sustain a considerable throughput, up to 86%, under the smart jammer. These results are a little different from the numerical analysis of section 4.1. In Fig. 6 of section 4.1, the throughputs of the channel hopping scheme and AJ-MAC are approximately 55% and 85%, respectively, at DW 20 (100 ms= $\alpha \times t$ = 20×5 ms), while they are about 67% and 86% in the simulation results. The time γt (summation of jamming detection, announcement, and switching time) and jammer model differ slightly between the numerical analysis and the simulation, which may explain the difference. Fig. 9 shows the accumulated number of jammer's detection times for different channel hopping schemes. We can indirectly check that AJ-MAC has a lower jammer's detection probability than do the other two schemes. Fig. 10 shows the throughputs against the jam time. We can find that the impact on the throughput for a different jam time is not so critical in channel hopping schemes. Simulation results show AJ-MAC can overcome the defects of the prior research by increasing the throughput and decreasing the jammer's detection probability under the smart jammer.

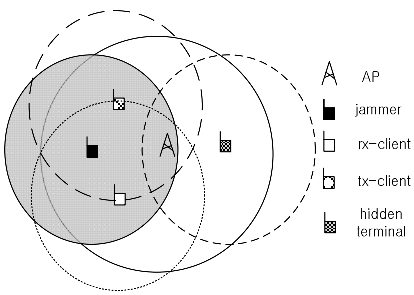


Fig. 7. Simulation topology

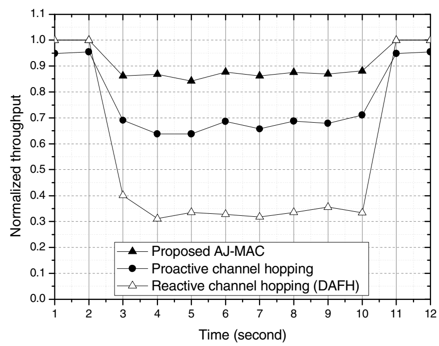


Fig. 8. Throughput for different channel hopping schemes

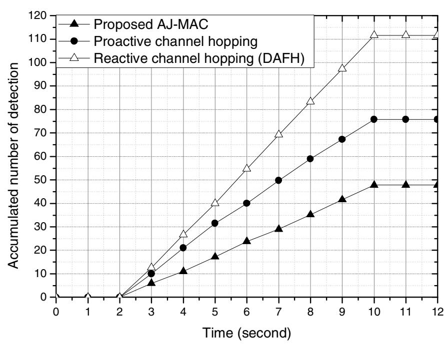


Fig. 9. Accumulated number of jammer's detection times for different channel hopping schemes

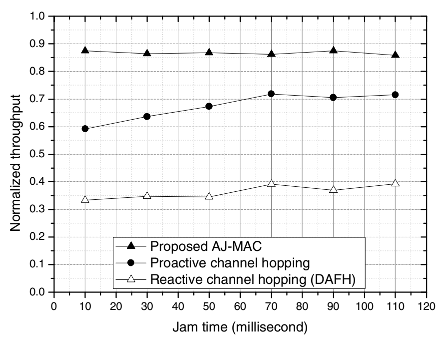


Fig. 10. Throughput for different channel hopping schemes based on jam time. The other parameters are the same as in Table 1.

5 Conclusions

In this paper, we introduced the Anti Jamming–based Medium Access Control (AJ-MAC) to mitigate physical layer jamming, especially a smart jamming in 802.11 systems. Using an adaptive rapid channel hopping scheme, AJ-MAC can overcome the defects of prior research: throughput degradation of proactive rapid channel hopping and high detection probability of a reactive channel hopping. AJ-MAC can achieve adaptive rapid channel hopping using the novel concepts of Dwell Window (DW) and Deception Mechanisms. The basic concept of DW is that a wireless network adaptively adjusts transmission time based on a jammer's ability. The basic concept of the Deception Mechanism is that the jammed transmitting-client deceives the jammer with a retransmission and makes the jammer attack an unnecessary channel. Numerical analysis and simulation results show that our proposed AJ-MAC is more effective than those of prior research.

In the near future, we will study anti-jamming schemes to overcome a more sophisticated smart jammer who can estimate the next hopping channel, such as a compromised node. Finally, we hope that our research will help 802.11 systems be more robust.

References

1. Ramakrishna, G., David, W., Ben, G., Srinivasan, S.: Understanding and mitigating the impact of RF interference on 802.11 networks. In: Proc. of SIGCOMM 2007, vol. 37, pp. 385–396 (2007)
2. Vishnu, N., Aniruddha, B., Samrat, G., Dan, R.: Using channel hopping to increase 802.11 re-silience to jamming attacks. In: Proc. of 26th IEEE INFOCOM Minisymposium (2007)
3. IEEE Standard 802.11h, Amendment to part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification (2003)
4. Peter, P., Hiroyuki, Y., Ramjee, P.: Strategies for adaptive frequency hopping in the unlicensed bands. IEEE Wireless Communications Magazine 13, 60–67 (2006)
5. Jeung, J., Jeong, S., Lim, J.: Authentication – based medium access control to prevent protocol jamming: A-MAC. In: Taniar, D., Gervasi, O., Murgante, B., Pardede, E., Apduhan, B.O. (eds.) ICCSA 2010. LNCS, vol. 6018, pp. 72–84. Springer, Heidelberg (2010)
6. Gast, M.S.: 802.11 Wireless Networks: The Definitive Guide, pp. 32–66. O'Reilly Publisher, Sebastopol (2002)
7. Anmol, S., Christian, D., Dirk, G., Richard, H., Douglas, S.: MOJO: A Distributed Physical Layer Anomaly Detection System for 802.11 WLANs. In: Proc. of ACM MobiSys, pp. 191–204 (2006)
8. Jaemin, J., Seungmyeong, J., Jaesung, L.: A Method to Detect and Mitigate PHY and MAC layer Jamming for 802.11 systems. In: Proc. of ICONI 2010, Cebu, Philippines (December 2010)

Mobile Based HIGHT Encryption for Secure Biometric Information Transfer of USN Remote Patient Monitoring System

Young-Hyuk Kim, Il-Kown Lim, and Jae-Kwang Lee*

Hannam Univ., Department of Computer Engineering,
Daejeon, Korea
{yhkim,iklim,jklee}@netwk.hannam.ac.kr

Abstract. In the future, advanced countries have expected that will face serious social problems such as public health, medical, welfare. The increase of chronic disease caused by would lead to increase in medical costs, and also the national financial burden, under this circumstances, would be largely. Therefore, the field of u-healthcare attracts the world's attention, and advanced countries are supporting this u-healthcare project through(with) medical law revision. u-RPMS(USN Remote Patient Monitoring System) which study in this paper has emerged a new technology in the field of u-healthcare. The medical system using USN is aiming to send the data, which collected through body-mounted sensors of inside and outside, to the outside system which is able to monitor and manage the user, but there could be occurred any problems such as the private information leaks, loss and falsification. Therefore, we proposed biometric information security model used HIGHT encryption in order to solve these security vulnerabilities.

Keywords: USN, Mobility Monitoring, HIGHT, Biometric Information.

1 Introduction

U-Health, an abbreviation of ubiquitous health, is an integrated concept of Healthcare and Wellness. It can be defined as a service system or environment supplying health care services of prevention, diagnosis and post-management anytime and anywhere by combining medical industry and IT. U-Health is an ideal health care system to ensure human's healthy life by systematically connecting health care providers and beneficiaries[1]. U-Health divides telemedicine services roughly into telemedicine, remote monitoring, tele-consultation and remote support in terms of service types[2].

The demand for the health and health arising from the demand for well-being has been transformed to the demand for life extension as well as sustaining a happy and well-being life from the aspect of quality. The problem underneath such demand is that it cannot be provided to the subject who is mostly adjacent to the medical service. Patients with high risks and the old, who have drawn a lot of attention throughout the

* Corresponding author.

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 83–95, 2011.

world, are required to maintain their lives within the specific boundary that a hospital set[3]. Therefore, u-RPMS(USN Remote Patient Monitoring System), an ubiquitous health care system which allows monitoring health conditions of the old and patient with high risks who cannot access to medical benefits and convenient service, without any restriction on places, time and movements is investigated for the study.

Currently, a u- Health market has put its effort on executing remote control of a patient's disease. It shall lead to decrease unnecessary expenses of national health and medical budget along with medical expense burden on patients by preventing unnecessary outpatient visits and hospitalizations. The necessity of the medical system such as u-RPM is clearly supported by a dramatically increased percentage of the old[Table 1., Table 2.][Fig. 1.].

Table 1. Statistics Korea estimation, Annual population composition

Year	Male Population	65 age over	Percentage	Female Population	65 age over	Percentage
1960	**12,550,691**	**288,795**	**2.3**	**12,461,683**	**437,655**	**3.5**
1970	16,308,607	408,117	2.5	15,932,220	583,191	3.7
1980	19,235,736	544,568	2.8	18,888,039	911,465	48
1990	21,568,181	821,857	3.8	21,301,102	1,373,227	6.4
1995	22,705,329	986,341	4.3	22,387,662	1,670,311	7.5
2000	23,666,769	1,299,786	5.5	23,341,342	2,095,110	9.0
2001	23,843,136	1,382,713	5.8	23,514,226	2,195,657	9.3
2002	23,970,035	1,470,459	6.1	23,652,144	2,301,616	9.7
2003	24,089,703	1,559,346	6.5	23,769,608	2,408,772	10.1
2004	24,165,488	1,644,713	6.8	23,873,927	2,521,261	10.6
2005	24,190,906	1,733,661	7.2	23,947,171	2,632,981	11.0
2006	24,267,609	1,835,100	7.6	24,029,575	2,750,602	11.4
2007	24,344,276	1,938,638	8.0	24,112,093	2,871,725	11.9
2008	24,415,883	2,032,180	8.3	24,190,904	2,983,846	12.3
2009	24,481,480	2,112,786	8.6	24,265,213	3,079,924	12.7
2010	24,540,316	2,189,996	8.9	24,334,223	3,166,857	13.0
2011	**24,592,456**	**2,276,143**	**9.3**	**24,396,377**	**3,260,929**	**13.4**
2012	24,632,725	2,373,276	9.6	24,450,459	3,368,468	13.8
2013	24,665,686	2,478,334	10.0	24,497,130	3,483,891	14.2
2014	24,689,966	2,571,758	10.5	24,537,485	3,596,709	14.7
2015	24,706,848	2,678,037	10.8	24,570,246	3,702,782	15.1
2020	24,679,762	3,303,494	13.4	24,645,927	4,397,631	17.8
2025	**24,505,752**	**4,276,834**	**17.5**	**24,602,197**	**5,490,959**	**22.3**
2030	24,190,354	5,217,615	21.6	24,444,217	6,593,092	27.0
2040	22,854,325	6,647,579	29.1	23,488,692	8,393,328	35.7
2050	**20,734,181**	**7,131,572**	**34.4**	**21,608,588**	**9,024,185**	**41.8**

Table 2. Health expenditure compared to GDP(%)

Year	USA	England	German	France	Japan	**Korea**
1997	13.1	6.8	10.2	10.2	7.0	**4.2**
2006	15.3	8.4	10.6	11.0	8.1	**6.4**

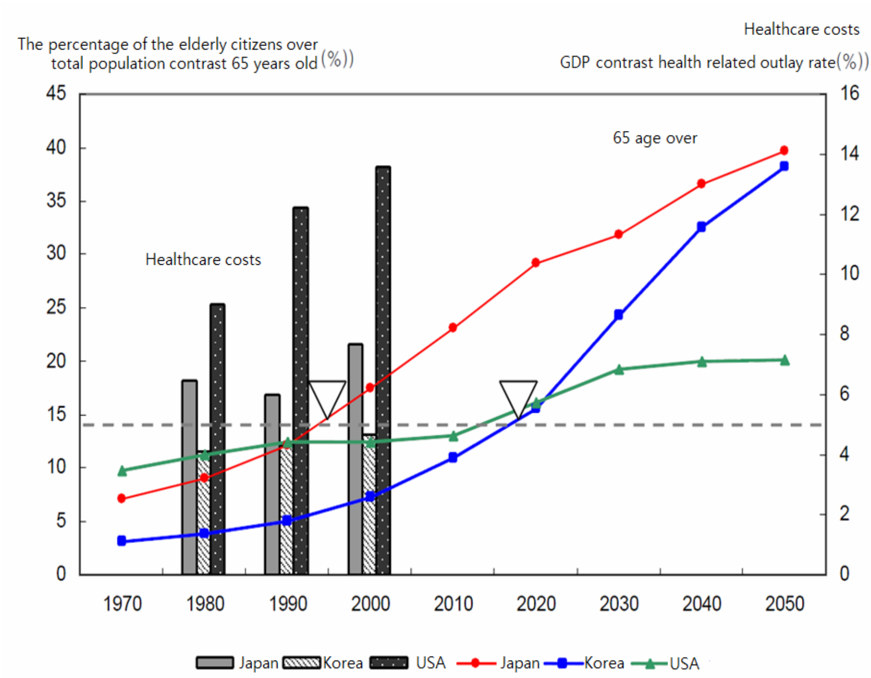


Fig. 1. Korea, Japan, US Aging information & Healthcare outlay and forecasts

Despite its well-disputed advantages, u-RPMS has a number of security issues since it collects and transmits individuals' medical information in an ubiquitous environment. The security issues can be roughly classified into 4 types.

First, a sensor in charge of collecting biological information is also in charge of managing and transmitting the information without any limitation on time. If unencrypted information was transmitted apparently, it could be modified under the malicious purpose to give a wrong prescription.

Second, as the number of a sensor node for collecting various biological information increases, routes to be protected are complicated and diversified which are obstacles for anticipation of possible attacks.

Third, a sensor, a representative device of u-Health is sensitive to a battery due to miniaturization. As for sensors inserting underneath the epidermis, batteries to be used must be semi-permanent. Nevertheless, if there is any attack aiming to consume battery reserve resources in concentrate, it might lead to unexpected interruption of overall services. For the last, the ongoing negative point of a system which transmits collected biological information to the exterior is that it always requires reporting the location of the subject, violating privacy. Even though a system is located far from the subject, it collects usual routes, habits and actions of the subject for 24 hours.

Considering these challenges, this study makes a proposal of using HIGHT encryption algorithm to properly protect and transmit private information directly related to the life. HIGHT is an encryption algorithm developed to be used in USN, RFID, and embedded computing environment. For assessing its compatibility with u-RPMS, its data processing speed shall be compared with of other encryption algorithms and analyzed.

2 Related Studies

We have seen a lot of active both domestic and overseas studies and researches regarding U-Health market on governmental and private levels. For reason of high medical expenses and aging, the U.S.A, Japan and many developed nations of Europe have promoted u-Health and managed to execute in earnest. Therefore, a u-Health global market is expected to reach 245 billion USD in 2013 [Table 3.], increased from 105.8 billion USD in 2007, with 15.7% annual growth rate in an average[4].

Table 3. u-Health Market, in world (Million $)

	2007	2009	2011	2013	Average growth rate
u-Medical	304.8	418.1	532.9	705.0	15.0%
u-Silver	199.1	247.0	288.7	347.0	9.7%
u-Wellness	553.9	766.3	1,071.8	1,487.7	17.9%
Total	1,057.8	1,431.4	1,893.4	2,539.7	15.7%

2.1 u-Health Trend

U.S.: the U.S. government, which has been initiating policies supporting U-Health, established a telemedicine specialized organization, ATA(American Telemedicine Association) in 1993 and legislated a law for exchanging medical information, HIPPA(Health Insurance Portability and Accountability Act) in 1996 and federal telemedicine corporation, HIT(Health Information Technology) in 1997[5]. Private enterprises of ICT(Information & Communication Technology) have utilized them under partnership.

Sprint: has a strategy of GE Healthcare supplying equipment and Sprint supplying wireless network based on partnership with GE Healthcare.

Verizon: has Verizon Business and Wireless healthcare project teams among 3 projects, supported by marketing healthcare group.

Deuthche Telecom: supplies Home monitoring telemedicine service through partnership with Philips, a medical device specialized company.

GE Healthcare: GE's Technology as one area of business has made e-Health services. An e-Health aims at the market occupancy using the medical image / Information Technology as the word naming the health care IT service supporting the medical information alternating current through internet network and communication.

Microsoft: MS entered into the health care industry through the HealthVault Community Connect. The individual medical records including a connection, admission and discharge information, prescription, check result, and etc. can read in the website of the affiliated hospital of a patient[6].

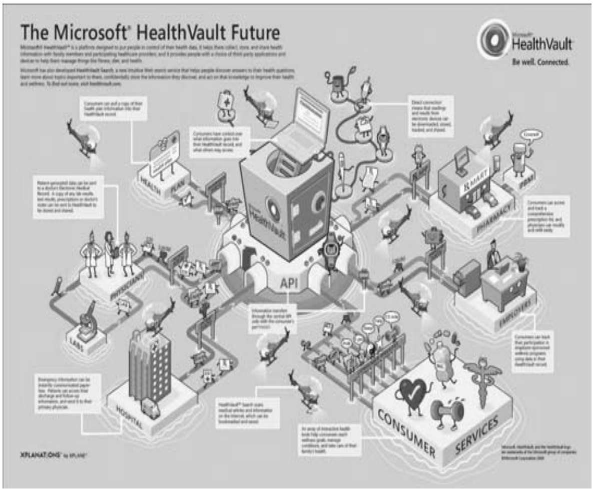


Fig. 3. Microsoft HealthVault Future

Japan: U-Health is referred as the same concept with telemedicine in Japan. Starting as CCTV and telephone wires in 1971 to provide medical care in back regions in Okayama-ken and being developed into remote support at disaster using communication satellite and ISDN in 1980s, telemedicine then expanded followed by MHLW(Ministry of Health, Labour and Welfare)'s regulation determining telemedicine using Info & communication equipment to conform with Doctors Act in Dec. 1997. The public's usage of telecommunication survey of MIAC(Ministry of Internal Affairs and Communications) in 2001 showed 43.2% of population used medical care through video screens[5].

NTT: NTT, a telecommunication company of Japan, has promoted the establishment of an intersystem based on digitalization of equipments, facilities and procedures of a hospital. In Oct. 2007, "wellness mobile phone" which can measure various biological information including body fat, pulse, breath smell and the number of walking steps, was co-developed with Mitsubishi Electronics. Most of businesses are focusing on informationization of medical service from the aspect of hardware.

Matsushita: "Good Sleep System" provides a pleasant sleeping condition by integrated controlling of 10 electronic devices, including lightings, an air conditioning system, and AV, was launched in 2006. In the future, a home health care system business is planning to be expanded as a comprehensive health care system including a service utilizing user's health information for a hospital and an office. There are anticipations of development of MICS (Medical Implant Communication System) which can monitor any defects within 24 hours without additional measurement, if technologies for a sensor reached a certain level[6].

Korea: with perspective of seeing U-Health as a representative integral new industry and strong drive to facilitate U-Health, the government presented 'Medical Act Amendment' approving telemedicine in April and proposed a bill of 'Health Care Service Law' allowing preventive health care based on personal health status in May in order to improve legal and institutional systems which have so far limited U-Health market.

In addition, SK Telecom consortium including Samsung Electronics and SNU(Seoul National Univ.) Hospital and LG consortium including LG Telecom and Gangnam Severance Hospital participated in 'Smart Care Pilot Project' organized by Ministry of Knowledge Economy[6].

Seoul city also introduced and invested in remote video emergency medical system as a 5 year project starting from 2006. The fire department of Pusan city has been already operating remote video first aid services as a part of U-119 emergency medical service since 2007 and Siheung city has dispatched ambulances applied with telemetry system in places patients of serious cases frequently transferred since Feb in 2010 while using HSDPA/WiBro for info transmission[7].

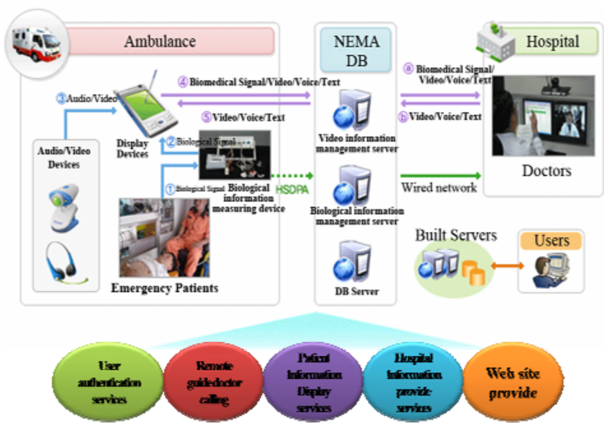


Fig. 4. Concept Diagram of Remote Video Aid Services in Pusan city

Fig. 5. Concept Diagram of Telemetry in Siheung city

USN based remote health monitoring system: This business project was officially chosen as the central business for RFID/USN expansion business in Dec. 2007, and in following April. 2004, business plans was determined and Gangneung-city(Gwangwon Province), Yeongyang-goon(Gyeongsangbook Province), and Boryeong-city(Choongchungnam Province) were decided to be the location for executing the business.

In case of Gwangwon Province, more than 80% of medical infra is concentrated in a city, and 87.5% of medical facilities are located in city and major towns in Choongchungnam Province. 12.5% of an area of the nation is classified as a township area including farming and fishing villages and mountain areas. As a blind spot for social welfare and medical services, these areas are deemed to be the medically undeveloed area. As for Gyeongsangbook Province, most of medical facilities are concentrated in certain major areas, such as Pohang city and Gyeongjoo-city, and there is no general hospital in districts. Yeongyang-goon, an area of 815km2 (1.3 times of an area of Seoul city), has a population of 19,696, and they are resided in scattered small villages, which result in a lot of troubles of accessing to a special hospital.

Table 4. USN based remote health monitoring system establish summary

Item	Contents
Name	USN based remote health monitoring system
Period	First intention: July 2008 ~ March 2009 Second intention: July 2009 ~ December 2009
Budget	Total 4.9 billion won Gangneung 10.55 billion Yeongyang-goon 10.61 billion Boryeong 12.21 billion Seosan 15.63 billion
Area	Application areas: four municipalities (Gangneung, Yeongyang-goon, Boryeong, Seosan) Participating institutions: health(medical) institutions 72, 90 doctors Services: telemedicine (video), u-visit nursing, health management, etc…

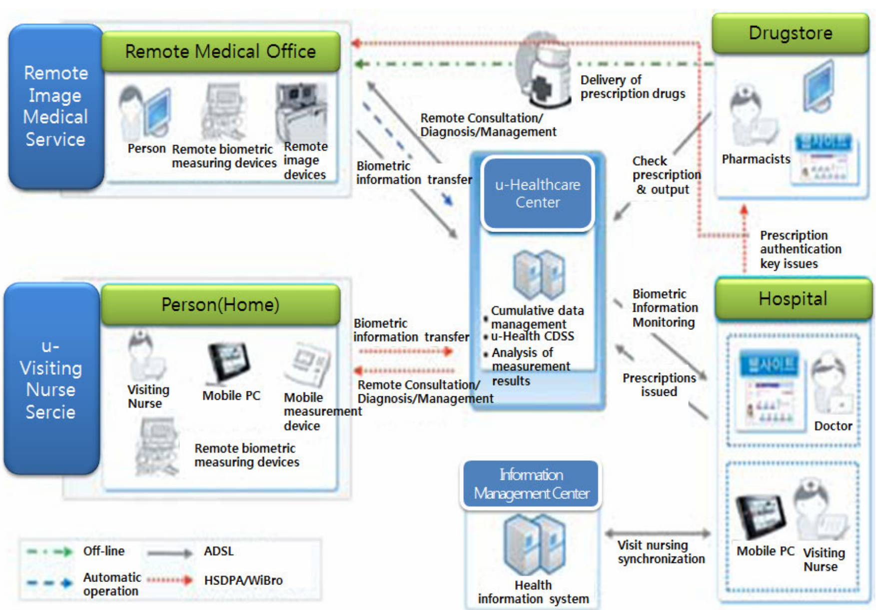


Fig. 6. USN based remote health monitoring system establish summary configuration

In April. 2009, Seosan-city(Choongchungnam province) was selected additionally. Major contents are listed in the [Table 4.], and [Fig. 6]. A remote health monitoring system shall be established to enhance a health medical service in a medically undeveloped area. The system links health maintenance organizations (a health center and a health support center) with a general hospital and a pharmacy in a neighboring city, with a community health post as a center. Then, telemedicine and electronic prescription, customized u-visiting nurse, and a home health care service for heart disease patients were provided.

3 u-RPMS Design

u-RPMS, a goal of this study, is commencing from integrating varied biological information collected by distributed each sensor node for u-Health service. [Fig. 7.] shows an overall structure of u-RPMS.

Each sensor node has different data loss rates, frequency rates, and data transmission delay rates due to dielectric substances, thickness, and impedance features varied upon locations of a sensor. Therefore, a medical organization, which must assess medical conditions of the subject by integrating distributed biological information, requires having the biological information collected during the same period for safety and reliability.

To fulfill these conditions, SG(Security Gateway) is installed and it collects and integrates information collected from distributed sensor node to formulate it to a single frame. Then, the information is encrypted to be transmitted to servers of a medical organization.

$$\mathrm{PL}(d) = \mathrm{PL}(d) + 10_{nlog}(d/d) + \mathrm{S}_{0100}\,Where\ (0)s\mathrm{SN}\sigma\mathrm{and}d50mm \quad (1)$$

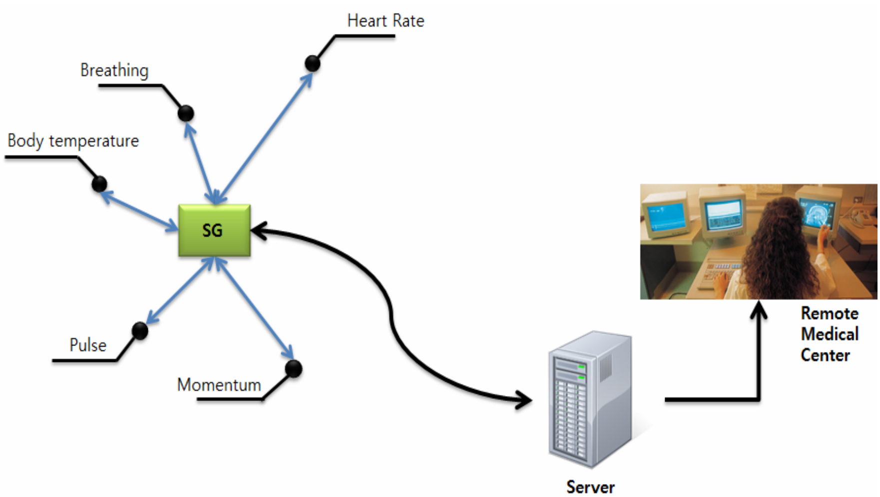


Fig. 7. u-RPMS configuration

Table 5. According to the statistical parameters of the path loss model

Human surface transplants	PL(d0)(dB)	N	σs(dB)
Deep Tissue	47.14	4.26	7.85
Epidermis	49.81	4.22	6.81

3.1 SG(Security Gateway)

Sensors installed at distributed locations to collect biological information transmit data to SG. One cycle is the time taking for a sensor to collect biological information, to transmit it to SG and to be ready for collect biological information again after the certain period for sleeping. Thus, each data transmitted to SG is counted and then, data from the same cycle is gathered to be transmitted to a medical organization.

3.2 Packaging

Due to features of biological information, formats of collected data are not unified. For the most basic format or the small size collection with one or two types, data can be collected with the basic formation of 'int' and later, '%' or 'C' can be added to a format of the data on the final stage. Despite of such efforts, it is inevitable that the number of biological gets increases and consequently of the data formation.

Therefore, data collected at SG is transformed to a 'String' format at once and the process is described in [Fig. 8.]. The payload transformed to a 'String' format uses the human body wireless network standard dada frame standard [8][9], and they are divided into each data and stored as it is at the final transmission stage with a separator of '-' as described in [Fig. 9.]. The payload created as the [Fig. 9.] is applied to the data frame structure of [Fig. 10.], and MSDU is transmitted to a PHY class to be PSDU.

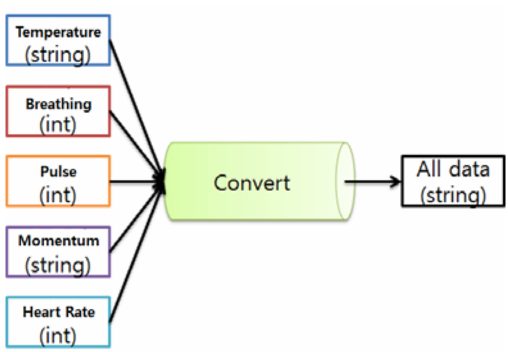


Fig. 8. Data type conversion

3.3 Encrypt

On this report, in order to provide confidentiality to the computing environment requires low electricity and light weight such as RFID and USN, HIGHT (HIGh security and light weigHT), 64 bit block encryption algorithm co-developed by National

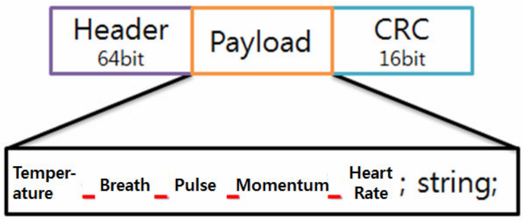


Fig. 9. Create payload using the separators

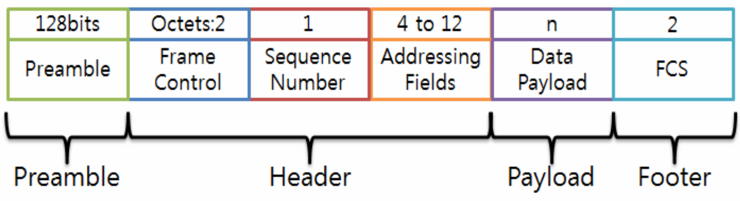


Fig. 10. Frame architecture

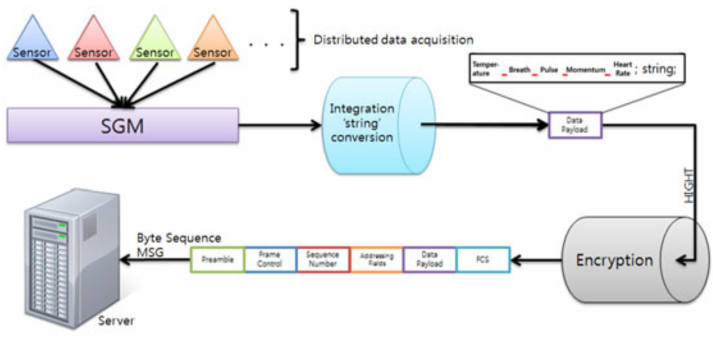


Fig. 11. Whole process

Security Research Center and Koryeo University is proposed to be used for data encryption. HIGHT is able to output 64 bit coded message from 128 bit master key and 64 bit plain text and has fast decoding and encoding time due to simpler structure than of SEED and AES with a low battery consumption rate[10]. [Fig. 11.] is the model proposed by this study, and an encryption process is added to aforementioned processes to transmit data to a server.

4 Experiments and Evaluation

4.1 Experiments

To conduct an assessment with aforementioned details, a simulated laboratory environment was realized. For the lab, 5 of simulated sensors were defined to process random numbers and to transmit them to SG.

SG was in charge of processing data (Packet, Frame, Header, Encryption and etc) and transmitting them to a server. A server received data and performed a role of DB control, and the simulation was executed by Net Framework 3.5/C#. HIGHT was implemented for encryption, and Windows 2008 Server and ubuntu 10 were used as a server. The overall system realization and the lab environment are revealed in the [Table 6.].

Table 6. u-RPMS implementation and simulation environment

Item	Contents
Language	C, C#/.Net Framework 3.5
Cryptography API	HIGHT encryption
Development Tool	Visual Studio 2008
Server	Intel Core2 Quad Q8400 2.66GHz, 4GB Intel Core2 Quad Q9400 2.66GHz, 4GB ARM9 S3C2440, SHT-75
OS	Windows Server 2008 Windows 7 64bit Ubuntu 10
DB	MySQL 5.1.41

As can be seen from [Fig. 12.], SG transmited the encrypted biological information to a server and the server decoded the encrypted biological information and stored it to DB using a separator.

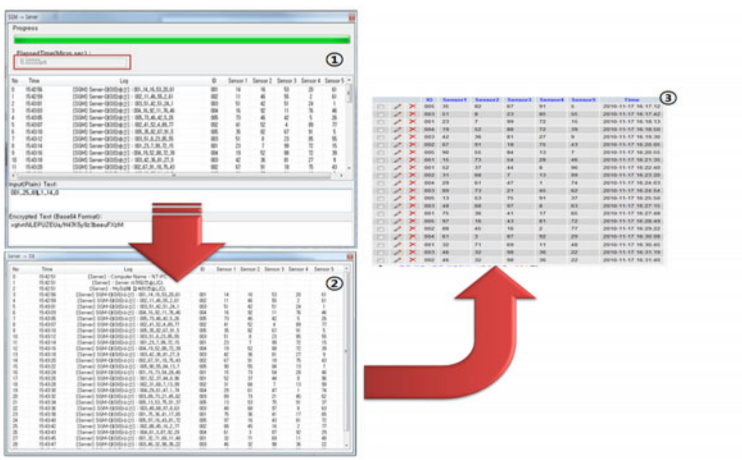


Fig. 12. u-RPMS Application

4.2 Evaluation

To confirm the time taking for transmitting biological information and whether there is any delay on transmission caused by HIGHT encryption, the performance of an encryption algorithm was assessed. [Fig. 13.] shows the time taking for that transmitted data from a sensor is transmitting to SG and then it is transmitted to a server finally and stored at DB before and after application of HIGHT.

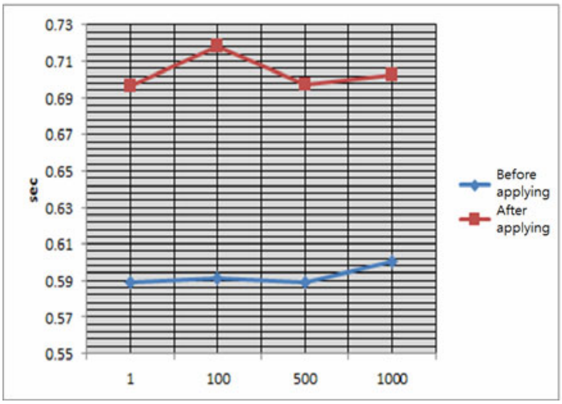


Fig. 13. Compared of Before & After applying

Sensor data was set to be created in an interval of 1 second. With the log and the time recorded after repeating it for the total of 1000 times, an average for 1 time, 100 times, 500 times and 1000 times were calculated. In general, there is no difference in a transmission delay because of fast encryption performance of HIGHT encryption. Usual performance of a hardware used for the device performing a role of SG for the other system has CUP performance of approximately 580 MHz. Thus, CPU performance was limited to 580 MHz for the experience and the speed was fixed to 1Mbps. For a temperature sensor, a sensing frequency with an interval of 2 to 3 seconds is recommended for thermal equilibrium. When a sensing frequency with an interval of 3 seconds for collecting biological information is assumed, around .1 second of a difference shall not make any significant influences.

5 Conclusion

The demand for the health starting from the demand for well-being has is not about simply extending the life. It has been developed into the demand for sustaining a happy and well-being life in anytime, anywhere and always. A fast transformation of the society into an aging society throughout the world shall lead to severe social problems in terms of public health, medical care and social welfare. Consequently, it will lead increased medical expenses along with increased burden on the national budget. Amid these circumstances, u-Health has gained a lot of attention and developed nations have supported u-Health while revising medical enactments.

On this study, u-RPMS which aims to monitor and manage the subject by transmitting the data collected by a sensor to the outside, was investigated. In case of a wireless environment, it has significant disadvantages. Information is prone to leakage, modification and loss in a wireless environment. Since u-RPMS handle handles information related to the life itself rather than simple individual's information, the way of managing and transmitting the collected data is very important issue. To resolve such challenges, a safety model to accurately and safely protect biological information has been developed. Each sensor distributed for collecting biological information was organized in a way that data collected in the same collecting cycle could be integrated into one. At SG, data with different formats is transformed to a 'string' format at once to be transformed to byte sequence data. This transformation is executed so the information transmitted to a server can be divided into original data to be easily stored at DB. This process also helps cope with delay in transmission and loss.

If information collected from a number of sensor was gathered as single coherent data, the number of count required to be attached on each sensor data can be decreased so as network consumption. Data created by aforementioned stages are encrypted by HIGHT, a national standard encryption algorithm, to formulate a frame and finishes its role by being transmitted to a server. A server acquires integrated data in a format of 'String' after decoding received data, and stored it to DB with a separator as a standard. This process results in storage of data in a format of character rather than the original data format. Moreover, information about a sensor including whether it collects temperature, humidity or acceleration is recorded, so when the data is displayed on the screen, all the additional information is added on the data automatically. As the conclusion of this study, HIGHT encryption which is light and effective in an environment and able to manage data fast can be applied to a device with various sizes and shapes and low arithmetic capacity to resolve security issues.

Acknowledgments. This paper was supported 2011 Hannam University Research Fund.

References

1. Park, S.-H.: Strateges for U-Health. GRI (2009)
2. Lee, J.-W., Kim, S.-H., Kim, C.-B., Kim, K.-K.: A Comparative Study on the telehealth regulations between U.S.A, Australia and Japan for developing the Korean telehealth system. Korean Journal of Medicine and Law 18(1) (2010)
3. Kim, Y.-R.: Comprehensive Study on Social and Legal Issues of Internet Medical Information. Media Science Research 10(2) (June 2010)
4. Jung, K.-H., Park, J.-H., Kim, J.-W.: U-health of the market towards the ICT industry realqualified War. DigiecoReports (2010)
5. Kang, M.-G., Kim, I.-K.: A Study on the Open STB based U-Health Services. KSII Journal 10(3) (2009)
6. Ok-Nam, K.: u-Healthcare is coming closer. HNFocus 26 (2009)
7. Park, S.-H.: Promotion Strategy of u-Health Business, Gyeongi Development Policies Research, 2009-32
8. TTAK.KO-10.0227: A method of transmitting PHY and Packet with an Interior of Human Body WBAN block base
9. TTAK.KO-10.0301: A structure of network physical stages of human communication network physical structure
10. Korea Internet Development Agency: HIGHT Block Encryption Algorithm Specification and Detailed Statement (2009)

Human Neck's Posture Measurement Using a 3-Axis Accelerometer Sensor

Soonmook Jeong*, Taehoun Song, Hyungmin Kim, Miyoung Kang, Keyho Kwon, and Jae Wook Jeon

School of Information and Communication Engineering, Sungkyunkwan University, Suwon, South Korea
{kuni80,thsong}@ece.skku.ac.kr, just_kidd@nate.com, river064@gmail.com, {kykwon,jwjeon}@yurim.skku.ac.kr

Abstract. Poor neck posture results in user strain. Prolonged use of computers is one of the major risk factors for neck pain in the workplace. This tendency is likely to continue as computer use increases. Such issues reduce a human's quality of life, and productivity. They can result in additional health care cost. Therefore, this research monitors the posture of a person's neck to help him/her avoid neck pain. We chose the LSODF (Least Squares Orthogonal Distance Fitting) to measure the user neck's posture in 3D geometry. This model fitting plays an important role in extracting geometric information from a set of given data points in 2D/3D space. Therefore, the LSODF needs a data set for model fitting. We chose to generate the data set using a 3-axis accelerometer. We intend to improve his/her neck posture via monitoring and feedback by this method.

Keywords: Neck pain, Neck posture monitoring, Accelerometer, Computer use, LSODF, Geometric information, Model fitting.

1 Introduction

Poor neck posture results in user strain. Prolonged use of a computer is a major risk factor for neck pain in the workplace [1]. Neck pain is a common complaint that affects nearly every individual at some point. Neck pain is generally a minor irritant that has little effect on an individual's everyday life. Whereas, severe or chronic neck pain can be debilitating. What are some of the common causes of neck pain? Most instances of neck pain are related to muscular strain or sprain.

Poor posture results in pain, muscular strain, and tension over time. Most severe disorders (neck, shoulder) are related to computer use in the workplace due to adopting positions of poor posture [2].

* This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency)" (NIPA-2010-(C1090-1021-0008))(NTIS-2010-(1415109527)).

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 96–109, 2011.

One recent prediction of computer use approximates that there were more than one billion computers in use at the end of 2008 [3]. This report also predicts a 12.3% compound annual growth rate between 2003 and 2015. Reports about the lifetime propagation of neck pain in the general population range from 67%~80% [4]. Research on 512 office workers found the 12-month propagation of neck pain to be 45.5% [5].

This social phenomenon comes from changing patterns of work required to perform their work more efficiently. Computer use is increasing as computers become more prevalent. Most schools and companies supply workers with a computer [6]. Workers and students have to implement a large part of their daily tasks on computer.

This trend is likely to continue, as computer use increases. The incidence of neck problems in computer users will be consistently found to be high. Such problems reduce a human's quality of life and productivity. They can result in additional health care costs [7]. Thus, we can expect a corresponding increase in the propagation of neck pain as computer use grow, if appropriate countermeasures are not prepared. Therefore, prior detection via neck posture monitoring is essential. This research monitors the user's neck posture to prevent neck pain.

Contribution. Neck flexion is a complex mechanism, as there are eight joints involved in head/neck flexion [8]. This means a human neck can generate complex geometric postures. However, previous research proposes measurement methods for specific posture that are too simple to detect complex neck movements [9], [10].
Therefore, a more precise measurement method is needed. We tried to find a solution to measure complex neck postures. We chose to use LSODF (Least Squares Orthogonal Distance Fitting) [11].

This model fitting plays an important role in extracting geometric information from a set of given data points in 2D/3D space (point cloud, coordinate data set) with pattern recognition and computer vision. We propose a method to measure the human's neck posture in 3D geometry using this model fitting.

LSODF needs a data set for model fitting. We chose to generate the data set using the 3-axis accelerometer, since it has became an appropriate tool for use in wearable computer systems to detect and measure features of human movement [12]. The sensor is miniature, inexpensive and requires low power.

For these reasons, the accelerometer has been used extensively for the measurement of human movement [13] and is entirely suitable for monitoring posture. Thus, we adopt the 3-axis accelerometer to measure human neck posture.

We installed this sensor into the commonly found inner Bluetooth earphone. The user wears the earphone in the ear. Then, the 3-axis signals(x, y, z) are generated based on the user's neck posture from various directions and angles. They are represented as points in 3D space. LSODF implements model fitting with these points in 3D space. It measures the user neck's posture that results in neck pain.

We intend to help people avoid neck pain by monitoring their neck posture.

The remainder of this paper is organized as follows. Section 2 introduces related work. Section 3 describes the system design for our measurement method. In Section 4, we demonstrate experimental results. A conclusion is provided in Section 5.

2 Related Works

Many researchers have investigated how computers affect neck pain. They have found that increases in neck angle result in load and torque increases on the cervical spine and increased discomfort [14]. Posture monitoring systems have been proposed by [15], [16]. Their systems use computer vision to determine the user's activities through his/her posture.

Software programs are also available to help reduce a user's stress injuries. Most of these programs are designed to remind you to take breaks and perform exercises. While most of these programs track mouse and keyboard use, they cannot track posture changes.

An accelerometer based method, proposed by [9], found that the specific vertebras are the most mobile vertebras in the backbone and most prone to be affected by poor posture when the person uses a computer. Thus, they measured head and neck angle by placing an accelerometer at the vertebras C7, directly measuring the skull-vertebral angle. Feedback consisted of a color signal and simple beep when exceeding acceptable thresholds. However, this method cannot detect the diverse geometrical motion of a neck, because it only measures the angle of the neck with respect to the back. Neck flexion is a complex mechanism, as eight joints are involved in head/neck flexion. That is, it is difficult to detect the neck motion in detail.

Another sensor-based method was proposed by [10]. This method used FLS-01 Flex Sensors, instead of computer vision and an accelerometer to monitor human neck movement. Each sensor had its own position: one on the front, one on the left, and one on the back of the neck. However, this method simply measures the user's neck angle in four directions (front, back, left and right). It is inconvenient to wear the device attached with sensors. Moreover, the device is connected to the PC by wire. Fig. 1 shows the common neck pain of everyday life.

Fig. 1. Neck pain of everyday life

3 System Design

We chose to use an accelerometer to measure the user's neck posture. Unlike previous methods, in which the sensor has mostly been placed around the neck, we attached the sensor to the ear. We use the Bluetooth earphone to attach and fix the sensor to the ear. One accelerometer is installed into the Bluetooth earphone and the user wears it on his/her ear. After the device is connected to the PC in Blue-tooth communication, it transmits the sensor signal, based on the user's head movement, to the PC. Then, the machine analyzes the signal and detects the neck posture using the proposed algorithm.

3.1 Prototype Device Design

The accelerometer sensor is installed into the inner Bluetooth earphone. The sensor signals are sampled using the microcontroller board and are transmitted via Bluetooth to the user's PC UART. The microcontroller board consists of the Atmega128 controller and myBluetooth-EX Module. Table 1 shows the components of our prototype. The prototype is segmented broadly into three modules: 1) Bluetooth earphone module, where the accelerometer is installed. 2) Microcontroller board that converts the analog signal into a digital signal and transmits the converted signal to PC UART via Bluetooth. 3) Battery module that provides 5V/1A. The left side of Fig. 2 shows three modules in order, and the right side shows the combined form of the modules. Then, the user wears the microcontroller board and accelerometer earphone on his/her right arm and right ear, respectively. Fig. 3 shows the user's appearance wearing the prototype device. The 3-axis acceleration signal is generated based on the user's head movement.

Table 1. Parts of the prototype device

Part Type	Model Description	Model Image
Microcontroller	ATMega128 16AU AVR Module (AM-128RAM)	
Accelerometer	3 Axis Acceleration Sensor Module (AM-3AXIS Ver.2)	
Bluetooth	myBluetooth-EX Module (115.2kB)	
Earphone	Samsung WEP-460 Bluetooth	

Fig. 2. The prototype device for the neck's posture detection

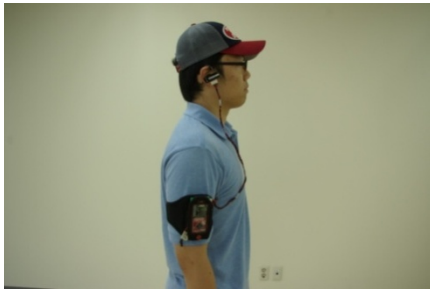


Fig. 3. The appearance wearing the prototype device

3.2 Algorithm

We chose to use LSODF to measure the user neck's posture. Model fitting to a set of measurement data is the most frequently performed elementary task in many science and engineering disciplines. Model fitting, along with pattern recognition and computer/machine vision, plays an important role in extracting geometric information from objects taken in a 2D image or 3D range image containing the curve/surface structures of the objects. LSODF implements the model fitting to a set of given data points in 2D/3D space (point cloud, coordinate data set).

We check the general posture related to the human neck, before measuring neck posture. Fig. 4 shows the five general types of posture.

Except for the posture (a) of Fig. 4, each posture has direction. While the postures (b), (c) are measured by previous methods, the postures (d), (e) are difficult to measure using them. This paper proposes a method to measure from posture (a) to posture (e).

First, we adopt the line fitting of the model fitting methods to measure postures (b), (c), and (d). This method generates the straight-line equation for neck movements, because those postures are mainly a straight motion. Using this equation, we measure the direction and velocity of neck movements and determine the neck posture. We adopt circle/sphere fitting to measure posture (e). This method generates the circle/sphere equation of the neck movements, because this posture is mainly rotation motion. The following subsections describe the process to generate those equations from a 3D point set.

(a) (b) (c) (d) (e)

Fig. 4. General postures of human neck (a) Normal pose (b) Neck inclination back and forth (c) Neck inclination towards shoulder (d) Neck turning right and left (e) Neck rotation

3.2.1 Line Fitting for Neck Posture Measurement

The line containing the point X_o and parallel to the unit vector r in n-dimensional space ($n \geq 2$) can be described in parametric form below:

$$X_o + ur \qquad \text{with} \qquad \| r \| = 1 \qquad \text{and} \qquad -\geq \ \geq u \geq \gg \ggg \tag{1}$$

Fig. 5 shows the straight line that most accurately represents a given point set and the d_i. d_i denotes the error between X_i and X_{ci}. Therefore, we should find the line that minimizes this error distance.

The square sum of the orthogonal distances from each given point $\{X_i\}_{i=1\sim m}$ to the equation (1) is

$$\sigma^2 \cong \sum_{i=1}^{m} ||(X_i - X_o) \times r||^2 \gg \tag{2}$$

Given r, X_o is looked for that minimizes (2), e.g. for n = 3,

$$\frac{\partial \sigma_o{}^2}{\partial X_o} = -2\left(\sum_{i=1}^{m}(X_i - X_o),\ \sum_{i=1}^{m}(Y_i - Y_o),\ \sum_{i=1}^{m}(Z_i - Z_o)\right)(I - rr^T) = \geq 0^T \gtrsim \tag{3}$$

There is only a trivial solution to (3) with r of arbitrary direction not parallel to any axis of the coordinate frame XYZ.

$$\sum_{i=1}^{m}(X_i - X_o),\ \sum_{i=1}^{m}(Y_i - Y_o),\ \sum_{i=1}^{m}(Z_i - Z_o) = 0 \tag{4}$$

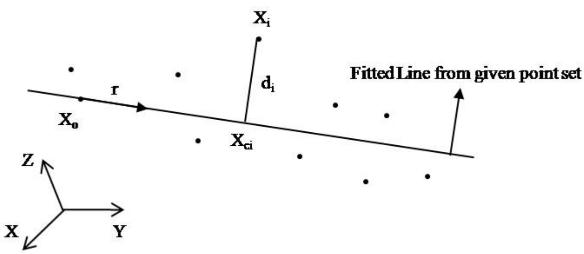


Fig. 5. Line fitting from given points $\{X_i\}_{i=1\sim m}$

From (4), we obtain

$$X_o = X = \frac{1}{m}\sum_{i=1}^{m} X_i,\ Y_o = Y = \frac{1}{m}\sum_{i=1}^{m} Y_i,\ Z_o = Z = \frac{1}{m}\sum_{i=1}^{m} Z_i \tag{5}$$

If r is parallel to one of the coordinate frame axes, the coordinate value of X_o may be set along this axis in an arbitrary fashion and the trivial solution (5) still applies.

If

$$x_i = X_i - X_o, \quad y_i = Y_i - Y_o, \quad z_i = Z_i - Z_o,$$

$$M_{xx} = \sum_{i=1}^{m} x_i^2 \ , \ M_{yy} = \sum_{i=1}^{m} y_i^2 \ , \ M_{zz} = \sum_{i=1}^{m} z_i^2 \ ,$$

$$M_{xy} = \sum_{i=1}^{m} x_i y_i \ , \ M_{yz} = \sum_{i=1}^{m} y_i z_i \ , \ M_{zx} = \sum_{i=1}^{m} z_i x_i$$

are set and the matrix H(inertia tensor of centroidal moment) defined

$$H \cong \begin{pmatrix} M_{yy}+M_{zz} & -M_{xy} & -M_{zx} \\ -M_{xy} & M_{zz}+M_{xx} & -M_{yz} \\ -M_{zx} & -M_{yz} & M_{xx}+M_{yy} \end{pmatrix} \tag{6}$$

The Square sum of the orthogonal distances (2) becomes

$$\sigma^2 = r^T H r$$

With the line (7)

$$X + ur$$

The symmetric square matrix H (6), extended for n-dimensional space ($n \geq 2$) without loss of generality, is decomposed by the SVD

$$H = V_H \, W_H V_H^T$$

With (8)

$$W_H = [\mathrm{diag}(w_{H1}, \ldots, w_{Hn})], \ V_H = (v_{H1}, \ldots, v_{Hn}), \text{ where } V_H^T V_H = I$$

The singular values $w_{H1}, \ldots, w_{Hn}$ and the orthogonal vectors $v_{H1}, \ldots, v_{Hn}$ are the principal centroidal moments and the principal axes of inertia, respectively. If r is parallel to one of the principal axes v_{H1}, from (7) - (9), we obtain (10): (9)

$$\sigma^2 = r^T H r = v_{Hj}^T V_H W_H V_H^T v_{Hj} = w_{Hj} \tag{10}$$

From this fact, when choosing the index j of the smallest wHj (9), the LSODF problem of line is solved. The resultant LSODF line (1) is

$$\mathbf{X} + \mathbf{u}\mathbf{v}_{\mathbf{Hj}}, \ \mathbf{X} = \{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$$

In summary, the LSODF line for the **m** given points $\{X_i\}_{i=1\sim m}$ in n-dimensional space ($n \geq 2$) passes through the mass center (5) and is parallel to one of the principal axes of inertia associated with the smallest principal centroidal moment.

3.2.2 Circle/Sphere Fitting for Neck Posture Measurement

This section describes how to calculate the model coefficient that minimizes square sum of error between the model value and observed value. Fig. 6 shows the relationship between the circle model and observed data set. The circle/sphere model is expressed by the following equation.

$$f(X, Y, Z, A, B, C, D) = X^2 + Y^2 + Z^2 + 2AX + 2BY + 2CZ + D = 0$$

If

$$f(X, Y, Z, A, B, C, D) \neq 0$$

The observed value (X_i, Y_i, Z_i) does not exist on the circle/sphere.

$$Q = \sum_i f(X_i, Y_i, Z_i, A, B, C, D)^2 \tag{11}$$

Equation (11) is the square sum of error. This value should be minimized for most appropriate model fitting from a given 3D point set.

For A, B, C, D, differentiate Q using partial differentiation.

$$\frac{\partial Q}{\partial D} = 2\sum_i (X_i^2 + Y_i^2 + Z_i^2 + 2AX_i + 2BY_i + 2CZ_i + D) = 0$$

$$\frac{\partial Q}{\partial A} = 2\sum_i 2X_i(X_i^2 + Y_i^2 + Z_i^2 + 2AX_i + 2BY_i + 2CZ_i + D) = 0$$

$$\frac{\partial Q}{\partial B} = 2\sum_i 2Y_i(X_i^2 + Y_i^2 + Z_i^2 + 2AX_i + 2BY_i + 2CZ_i + D) = 0$$

$$\frac{\partial Q}{\partial C} = 2\sum_i 2Z_i(X_i^2 + Y_i^2 + Z_i^2 + 2AX_i + 2BY_i + 2CZ_i + D) = 0 \tag{12}$$

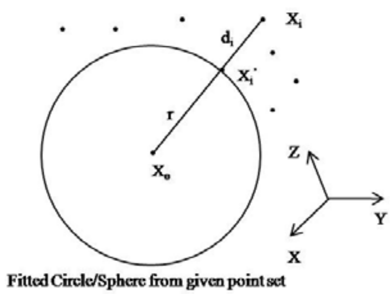


Fig. 6. Circle/sphere fitting from given points $\{X_i\}_{i=1\sim m}$

The equation (12) can be expressed as follows.

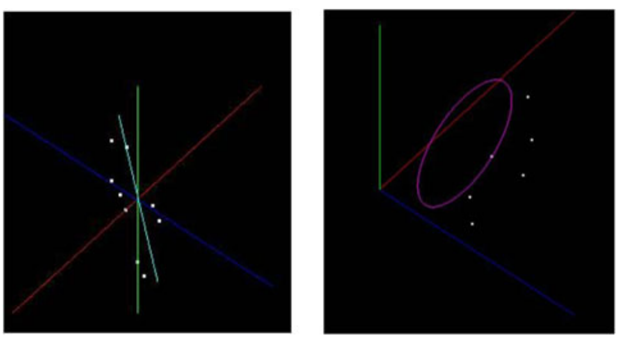
$$\begin{pmatrix} N & \sum 2X_i & \sum 2Y_i & \sum 2Z_i \\ \sum 2X_i & \sum 4X_i^2 & \sum 4X_iY_i & \sum 4X_iZ_i \\ \sum 2Y_i & \sum 4X_iY_i & \sum 4Y_i^2 & \sum 4Y_iZ_i \\ \sum 2Z_i & \sum 4X_iZ_i & \sum 2Y_iZ_i & \sum 4Z_i^2 \end{pmatrix} \begin{pmatrix} D \\ A \\ B \\ C \end{pmatrix}$$

$$= -\begin{pmatrix} \sum(X_i^2+Y_i^2+Z_i^2) \\ \sum 2X_i(X_i^2+Y_i^2+Z_i^2) \\ \sum 2Y_i(X_i^2+Y_i^2+Z_i^2) \\ \sum 2Z_i(X_i^2+Y_i^2+Z_i^2) \end{pmatrix}$$

Therefore the coefficient (X_0, Y_0, Z_0, R) of the 3D circle/sphere model can be obtained by the following.

$$(X - X_0)^2 + (Y - Y_0)^2 + (Z - Z_0)^2 = R^2$$
$$X^2 + Y^2 + Z^2 - 2X_0X - 2Y_0Y - 2Z_0Z + X^2 + Y^2 + Z^2 - R^2 = 0$$
$$X^2 + Y^2 + Z^2 + 2AX + 2BY + 2CZ + D = 0$$
$$\mathbf{X_0 = -A, \quad Y_0 = -B, \quad Z_0 = -C, \qquad R = \sqrt{A^2+B^2+C^2-D}}$$

Fig. 7 shows the test demonstration of the line fitting and the sphere fitting from the given 2D/3D point set.



(a) (b)

Fig. 7. Demonstration of (a) Line Fitting (b) Sphere Fitting in 3D space

4 Experimental Results

In this section, we monitor and measure the general neck postures that a person can pose. A major symptom of the neck pain can be detected by analysis of these postures. People generally do neck exercises to relax their neck tension when they experience suffering from the strain. The various neck postures are generated at this time.

Application. We can measure the neck pain level via a combination of these neck postures or the current neck posture. Moreover, an appropriate exercise for neck pain prevention can be implemented using the proposed method.

It is possible to distinguish the specific neck postures, since the proposed method provides the different model equation based on each neck posture. This paper proposes a method to distinguish the various neck postures.

We use the accelerometer data to detect the neck posture, as previously stated. However, in general, the accelerometer occasionally generates noise data, such as the error. This can cause the erroneous conclusion to analyze a pattern from the data set. Therefore, such a data set needs preprocessing to reduce the error data. This increases the cost, whether the preprocessing is implemented by software or hardware.

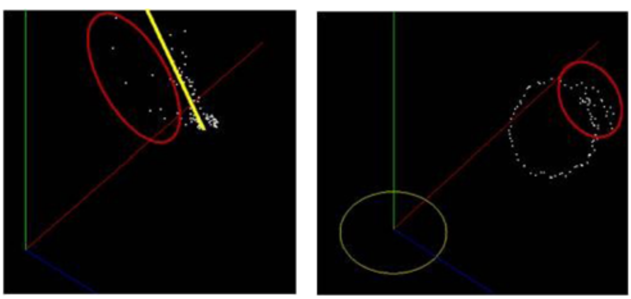
However, LSODF can solve such a problem, without preprocessing. The method finds the optimal model that most represents the given data set overall, although the data set includes some error data. Fig. 8 shows the generated model from the data set, including the error data using LSODF. The white dots represent the data set generated from the accelerometer; the data set in the red ellipse represents the error values. As shown in Fig. 8, the models (line, circle) are correctly generated, regardless of the error data set. We developed software using the OpenGL library with Visual studio 2008 to draw the figures and the accelerometer data set in 3D space.

For the generated line equation,

$$x = 534.167 + 23824944.510u,$$
$$y = 419.675 + 119123301.276u,$$
$$z = 329.822 + 156603061.591u$$

For the generated sphere equation

$$x = 362.111, \quad y = 408.606, \quad z = 394.324, \quad R = 179.641$$



(a) (b)

Fig. 8. Robustness of model fitting to accelerometer noise data: (a) Line Fitting (yellow line) (b) Sphere Fitting (yellow sphere, the center of it is moved to the origin coordinate)

We experimented if the proposed method measures the general postures. The general postures are classified into five types (1) normal posture (2) inclination back and forth (3) inclination towards shoulder (4) turning right and left (5) neck rotation. The following content demonstrates the experimental result. Fig. 9~Fig. 12 shows the line fitting based on each neck posture. The direction and length of the line determine the direction and angle of the neck movement.

1) Normal posture

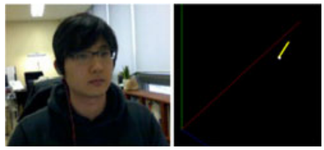


Fig. 9. Normal posture: most accelerometer data are concentrated on the small area (white dot)

2) Inclination back and forth

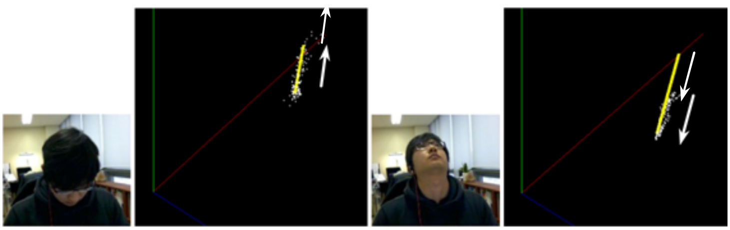


Fig. 10. Back and forth: the distribution of the data set is almost vertical

3) Inclination towards shoulder

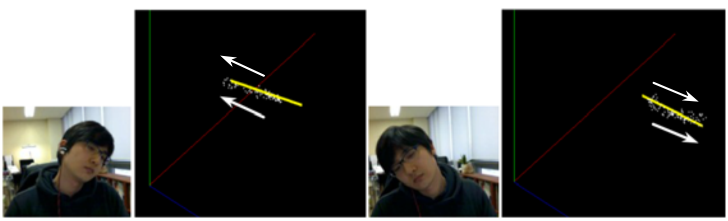


Fig. 11. Towards shoulder: the distribution of the data set is almost horizontal

4) Turning right and left

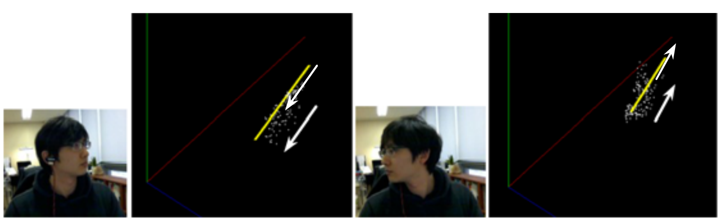


Fig. 12. Turning right and left: the distribution of the data set is almost parallel to the red axis

Fig. 13 shows a human's neck rotation. The sphere (yellow sphere) size is changed based on the size of the data set (white dots) generated from the neck movement. The sphere size represents the size of the neck rotation. Fig. 14 shows detecting the diagonal inclination of the neck. While it is not included in the typical postures, we add this demonstration to show our method can also measure these directions.

5) Neck rotation

Fig. 13. Neck rotation: the different sizes of the generated sphere are based on the sizes of the neck rotation (the center of sphere is moved to the origin coordinate of the 3-axis)

6) Neck's diagonal inclination

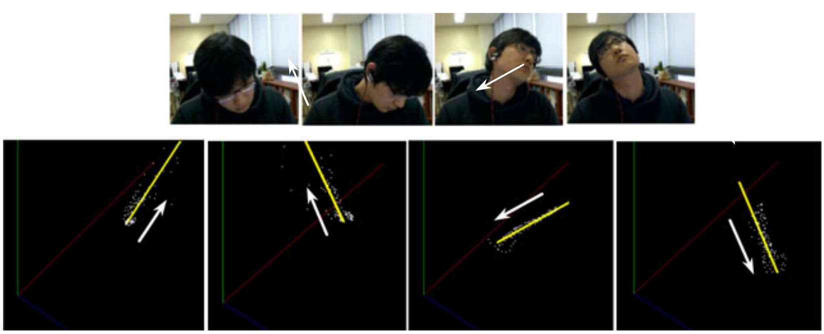


Fig. 14. Diagonal inclination of neck: each generated line has different direction

5 Conclusion

Poor posture causes human tension. Consistent use of a computer is a major risk factor for neck pain in the workplace. Therefore, this paper proposed a method to prevent workers who use a computer from suffering from neck pain. This method detects the major symptoms for neck pain, since it can measure the general neck posture.

However, we did not provide an evaluation method for user performance related to the neck pain in this paper. This paper proposes the method to be able to detect the various neck postures. Nevertheless, this method can be applied in various areas.

For example, the proposed method can be adopted in a self-exercise program to prevent neck pain. This program instructs the user to follow the motions related to neck exercises. He/she can then know whether he/she correctly followed the instructions. This is achieved by analyzing the generated model based on the user neck's posture. This means that the proposed method can distinguish specific neck postures.

We will apply our method to the application to evaluate the user performance in future work.

References

1. Fredriksson, K., Alfredsson, L., Thorbjornsson, C.B., Punnett, L., Toomingas, A., Torgen, M., Kilbom, A.: Risk Factors for Neck and Shoulder Disorders; a Nested Case-control Study Covering a 24-year Period. Am. J. Ind. Med. 38, 516–528 (2000)
2. Haufler, A.J., Feuerstein, M., Huang, G.D.: Job Stress, Upper Extremity Pain and Functional Limitations in Symptomatic Computer Users. Am. J. Ind. Med. 38, 507–515 (2000)
3. Yates, S., Daley, E., Gray, B., Gownder, J., Batiancila, R.: Worldwide PC Adoption Forecast, 2007 To 2015. Forrester Research (2007)
4. Green, B.: A Literature Review of Neck Pain Associated with Computer Use. Public Health Implications 52, 161–167 (2008)
5. Cagnie, B., Danneels, L., Van, D.: Individual and Work related Risk Factors for Neck Pain among Office Workers. A Cross Sectional Study 16, 679 (2007)
6. Straker, L., Jones, K.J., Miller, J.: A Comparison of the Postures assumed when using Laptop Computers and Desktop Computers. Applied Ergonomics 28, 263–268 (1997)
7. Mekhora, K., Liston, C.B., Nanthavanij, S., Cole, J.H.: The Effect of Ergonomic Intervention on Discomfort in Computer Users with Tension Neck Syndrome. Applied Ergonomics 26, 367–369 (2000)
8. Ankrum, K.J.N.D.R.: Head and Neck Posture at Computer Workstation-What's Neutral? Presented at 4th Triennial Congress of Internation Ergonomics Association (2000)
9. Paul, P., Breen, A.N., Gearoid, O.: Evaluation of a Single Accelerometer based Biofeedback System for Real-time Correction of Neck Posture in Computer Users. In: 31st Annual International Conference of the IEEE EMBS (2009)
10. Carrie, D., Sriram, S., Jon, T.: Posture Monitoring and Improvement for Laptop Use. In: CHI (2007)
11. Sung, J.A., Wolfgang, R., Hyung, S.C., Hans-Jurgen, W.: Orthogonal Distance Fitting of Implicit Curves and Surfaces. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(5) (2002)
12. James, F.K., Huw, W.B., Stamatina, A., Chris, B., Anthony, S., Theodoros, N.A.: Uses of Accelerometer Data Collected from a Wearable System. In: Pers. Ubiquit. Comput. (2007)

13. Lyons, G.M., Culhane, K.M., Hilton, D., Grace, P.A., Lyons, D.: A Description of an Accelerometer-based Mobility Monitoring Technique. Med. Eng. Phys. 27, 497–504 (2005)
14. Bonney, K.A., Corlett, E.N.: Head Posture and Loading of the Cervical Spine. Applied Ergonomics 33, 415–417 (2002)
15. Jaimes, A.: Posture and Activity Silhouettes for Self-Reporting, Interruption Management, and Attentive Interfaces. In: Proceedings of IUI 2006, pp. 24–31. ACM Press, New York (2006)
16. Jaimes, A., Liu, J.: Sit Straight (and tell me what I did today); A Human Posture Alarm and Activity Summarization System. In: Proceedings of CARPE 2005, pp. 23–34. ACM Press, New York (2005)

A System Consisting of Off-Chip Based Microprocessor and FPGA Interface for Human-Robot Interaction Applications

Tae Houn Song, Soon Mook Jeong, Seung Hun Jin, Dong Kyun Kim, Key Ho Kwon, and Jae Wook Jeon

School of Information and Communication Engineering, Sungkyunkwan University, 440-760, Suwon, Korea
{thsong,kuni80,coredev,bluster}@ece.skku.ac.kr, {khkwon,jwjeon}@yurim.skku.ac.kr

Abstract. In this paper, we describe off-chip based microprocessor and interface between FPGA systems. The off-chip process can reduce the time, cost, and complexity of system design and development as compared to systems based on programmable chips. This paper describes the structure of the Off-chip based interface system, communication of multiple-processor, communication of FPGA-Microprocessor and lastly the process of application development. In addition, the performance of Off-chip based interface system is verified by applying developed application to Autonomous Guided Vehicle.

Keywords: Multimodal, System Consisting, Off-chip, Microprocessor, FPGA, Autonomous Guided Vehicle Applications.

1 Introduction

1.1 Motivation

The microprocessor appeared in the year 1970 and has been used as an essential tool for developing sensor interfaces, path planning and intelligent algorithms, which are required for robots, by utilizing various sequential programming languages such as assembly language, C and C++. In the recent ten years, FPGA (Field Programmable Gate Array) that can parallel process large amounts of data in real time has been actively utilized in robots to enable them to communicate more naturally with humans, and to enable such robots to offer various services. Present research continues to make progress in developing a System on a Programmable Chip (SoPC) based robot-system that can accommodate all the merits of FPGA and microprocessors. However, the license cost for developing a SoPC system is extremely high compared to that of developing a microprocessor or FPGA; moreover, SoPC system is complex better than the other integrated system, and it is limited by universalized preceding study cases on development.

The off-chip based microprocessor and FPGA interface system proposed in this paper can simplify and improve the development process compared to that of

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 110–119, 2011.

developing the SoPC based system. Furthermore, the proposed system effectively reduces the cost of the development license.

1.2 Related Work

Human interest in robots has been increased because of the influence of movies, TV and through all sorts of multimedia. Consequently, people are more interested in robots that can offer convenient services in daily living rather than seeing these robots substitute simple labor in factories or other danger zones. Robots must communicate more naturally, and various systems must be developed in order for the robots to interact with and offer more convenient services to humans. To this end, progress continues in research on FPGA (Field Programmable Gate Array) that can parallel process great amounts of data in real time by utilizing existing microprocessors.

There is research on controlling the Car-like Autonomous Guided Vehicle via movement similar to human movement [1]. There is also research on materializing evolutionary algorithm and action controller based on FPGA, on controlling Autonomous Guided Vehicle [2]-[3], and lastly on materializing the FPGA base with the motor controller used inside the robot that cooperates with humans [4]. FPGA has been utilized to perform specific objectives in the latter research. However, altering the utilization objective of FPGA or utilizing it to add the application in a short period of time requires tremendous time and effort.

Studies on effectively developing various applications utilized in FPGA and processors include 1) an effective progress method of embedded product plan by utilizing FPGA and internal processor core [5]; 2) designing a Multiprocessor System-on-Chips that implements the ripple effect in general communication, multimedia, networking and many other studies [6]; 3) studies on guidance, navigation and control of Unmanned Aerial Vehicles utilized in DSP (a floating point Digital Signal Processor) and FPGA [7]; 4) a method to effectively advance the design of memory embedded in products by utilizing the internal processor core and FPGA [8].

However, the development method of inserting a processor core inside the FPGA to develop System on a Programmable Chip based system poses complexity due to requiring knowledge of computer structure, hardware and software. Furthermore, the Hard-core (Physical processor core that can be inserted inside the System on a Chip to interlink with FPGA) and Soft-core (Processor core composed with software by utilizing Hardware Description Language to be inserted inside the FPGA) used in SoPC have extremely high license cost and their development process is not as popular as that of Off-chip processors.

Section II describes the originally released low cost microprocessor and FPGA based interface system architecture, communication between microprocessor, and communication between microprocessor and FPGA. Section III introduces the application scenario of the Autonomous Guided Vehicle by utilizing the microprocessor and FPGA interface system. Lastly, the results and limitations of the Off-chip based robot system proposed in this paper are presented.

2 Off-Chip Based Microprocessor and FPGA Interface

2.1 Microprocessor and FPGA Interface System

The Off-chip based interface system consists of a Microprocessor Unit and a FPGA Unit. A total of five Atmel 8-bit Microcontroller Atmega128 (hereinafter "AVR128") are included in the Microprocessor Unit. One Master AVR and four Slave AVRs can be classified as five AVR128. Slaves 1-3 are connected with the FPGA Unit's FPGAs through the RS-232C communication protocol. Slave 0 AVR is connected to 4-Wheels NTREX NT-GIANTII [9] mobile robot via the RS-232C communication protocol. The Master AVR is connected to the RS-232C port of a Personal Computer used for debugging during system development, and can it exchange information with FPGA through Slaves 0-3 AVR and through a Serial Peripheral Interface (SPI). SPI allows for a high-speed synchronous data transfer between the ATmega128 and peripheral devices or between several AVR devices [10]. The FPGA Unit includes two Xilinx Virtex-4 XC4 VLX200 FPGAs (hereinafter "Virtex-4") and one Virtex-4 FX12 FPGA (hereinafter "Virtex-4"). The Virtex-4 included within the FPGA Unit has been embodied in each different function of Xilinx ISE Tools via VHSIC Hardware Description Language (VHDL).

The rest of RS-232C communication, besides the RS-232C for communication between Master AVR and Personal Computer, is connected via SENA SD202 Bluetooth RS-232C Serial Dongle [11] wireless communication [Fig. 1].

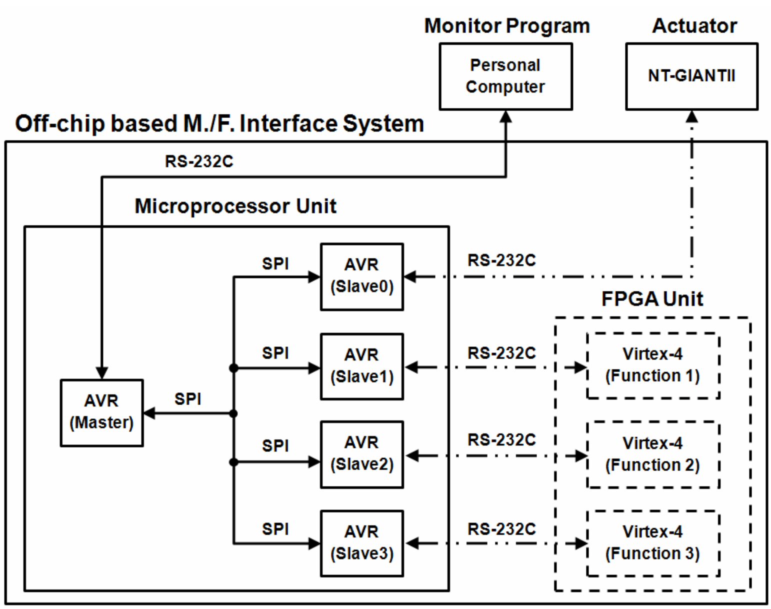


Fig. 1. Off-chip based microprocessor and FPGA interface system composition

2.2 Communication among Processor

Master AVR can control Slaves 0-3 AVR by utilizing SPI communication, and it can obtain data from the Slave AVR. SPI communication uses Slave Select (SS), System Clock (SCK), Master Input Slave Output (MISO) and Master Output Slave Input (MOSI) signals [Fig. 2]. The Slave AVR commonly utilizes the remaining signals such as SCK, MISO and MOSI in addition to SS_0, SS_1, SS_2 and SS_3 signals. Previous research has already verified its effectiveness in designing controls and communication architectures to realize embedded robot operating system for multi sensor interface system through SPI communication [12]. Slave code has been cloned in this research to effectively manage the numbers of Slave AVR codes; we have defined it as Slave Code Clone (SCC). The SCC provides methods to design the consistent input-output of Slave AVR hardware and to prepare one Slave code by later unifying the architecture of Slave code to identically utilize all the Slave AVRs. Therefore, managing the SCC code is not difficult even with an increased number of Slaves. The developer will see the effect of framing only two codes such as the Master and Slave AVR codes [Fig. 3].

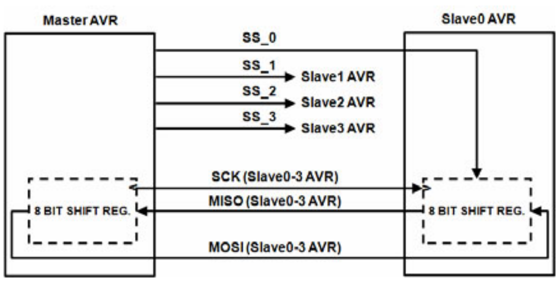


Fig. 2. SPI Communication Signal for Processor to Processor Interface

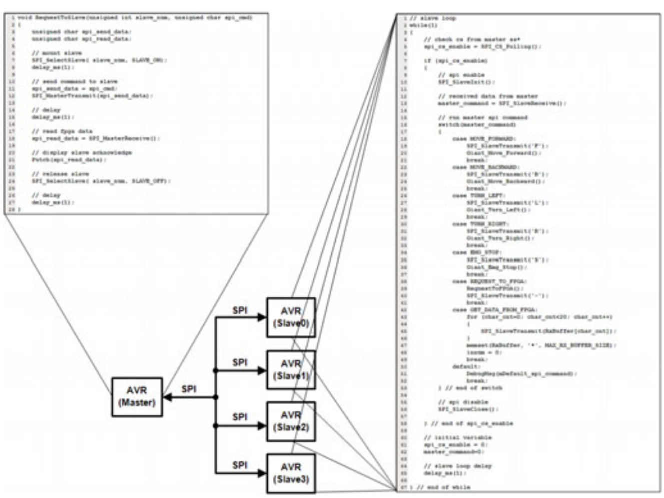


Fig. 3. The concept of Slave Code Clone

2.3 Microprocessor and FPGA's Communication Protocol

Development expediency of Off-chip based microprocessor and FPGA interface systems is very important. Various environments and variables must be realized and included in the development of Robot applications that provide users with convenient support that differs from developing an algorithm that is required for real-time parallel processing. Therefore, there are characteristics related to frequently adding and revising programs. Therefore, the location is proposed to separate the Microprocessor Unit and FPGA Unit installation space during the application development [Fig. 4]. There is need to establish developmental environment to offer appropriate service by the conditions and external interference generated during robot's navigation in offered application by the vehicle. Therefore, FPGA Unit functions with infrequent revisions are stationed in the robot.

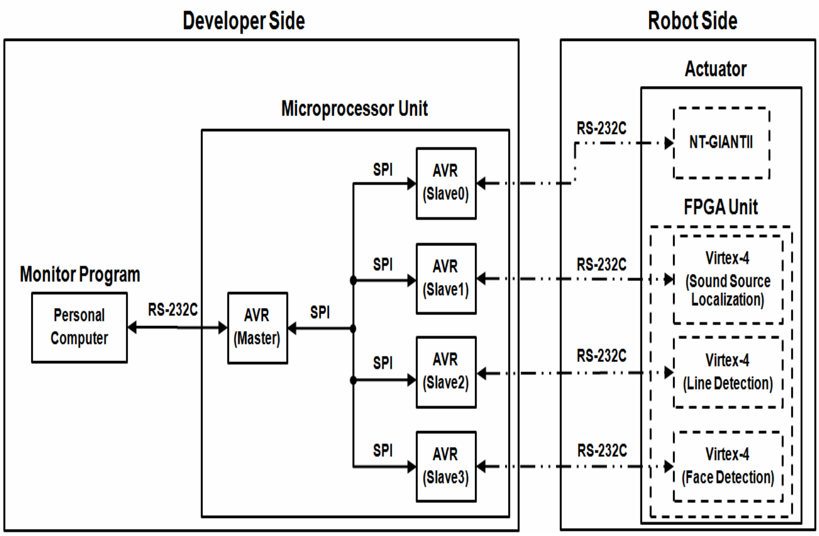


Fig. 4. System configuration by its installation location

On contrary, the Microprocessor Unit that demands frequent addition or revision will be stationed with the developer. Thus, RS-232C is suggested as a convenient method to interface between the microprocessor and FPGA under environment where installation space of microprocessor Unit in developer side and FPGA Unit in robot side remains separated. Not only RS-232C communication is widely used in microprocessor, but it also has an advantage that it can also be realized in FPGA. Furthermore, the RS-232C interface has an advantage of easily changing the environment into wireless by utilizing the Bluetooth Dongle. In this paper, RS-232C communication has been utilized to offer information required by applications such as FPGA based sound source localization [13], line detection [14] and face detection [15] where the development has already been completed.

Sound source localization, line detection and face detection communication protocol realized by microprocessor and FPGA are utilized by the ASCII based text. The basic architecture method of the protocol is to transmit the results that FPGA obtained (handled) between the Start text (“<”) and the End text (“>”). The FPGA system with sound source localization will transmit the bearing information with marking of -180 ~ +180 to application in 6Bytes text.

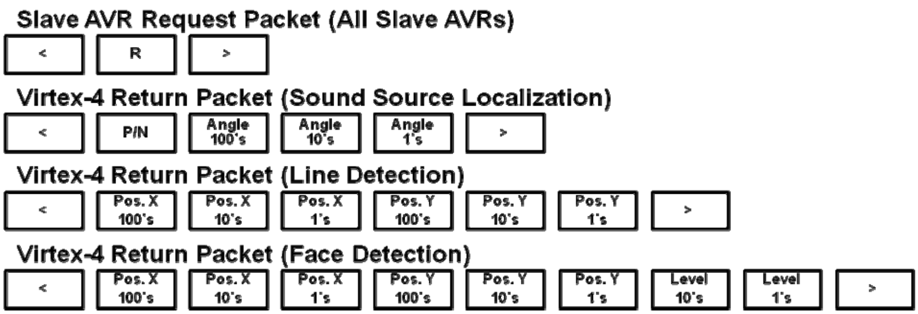


Fig. 5. RS-232C based communication protocol.

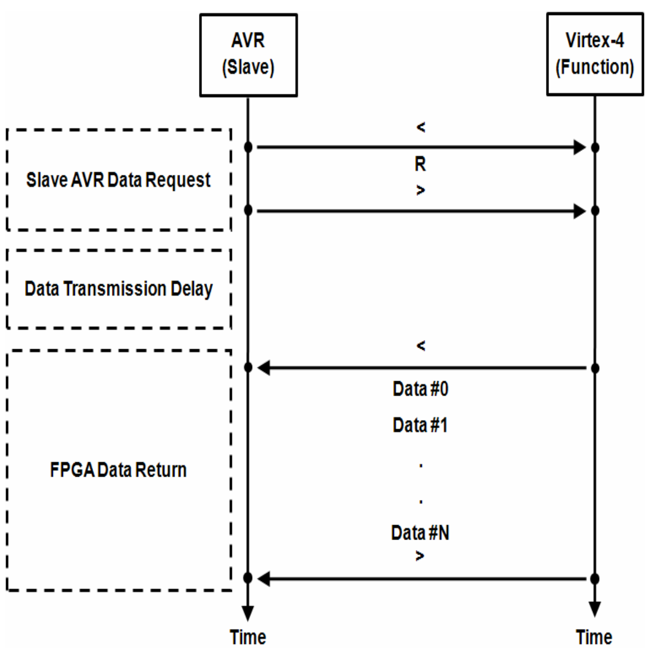


Fig. 6. Flow of Slave AVR and Virtex-4 FPGA interface

The FPGA with line detection will output the coordinates of the intersection of left/right of straight line existing on the route of the robot and will offer information to the application through 8Bytes of text. The face detection system will transmit the center coordinates of face closest to the image's Vertical Axis and the size of the face to the application through 10Bytes of text [Fig. 5].

To describe the communication current between the microprocessor and FPGA, the Master AVR requests information from the Slave AVR via SPI communication, and the Slave AVR will then transmit "<", "R", ">" to FPGA linked with RS-232C to request the processing results. FPGA that receives a request for data transmission will insert processed information between Start and End to send the data that FPGA has obtained via the Slave AVR [Fig 6]. For example, on the assumption that the result of sound source localization is +135, "<", "P", "1", "3", "5", ">" transmitted to the application and, in the case of line detection "<", "3", "2", "0", "2", "4", "0", ">", will be returned to the application when the point of intersection between the Right and Left of navigating floor is 320 and 240. Lastly in the case of face detection, "<", "3", "2", "1", "2", "0", "0", "0", "8", ">" will be returned to the application when the center coordinates of the face are 321, 200 and the size of the face is 8.

3 HRI Application Scenario and Results

Research on sound source localization [13], line detection [14] and face detection [15], was initially carried out by Jin Seung-hoon, Kim Dong-kyun and other researchers through FPGA base that has been used to frame the application through a microprocessor and FPGA interface system, proposed in this paper, and explained its application case. The application scenario is for the robot to usher the visitor of the laboratory to the Demo room and to perform human tracing through sound source localization and face detection [Fig. 7].

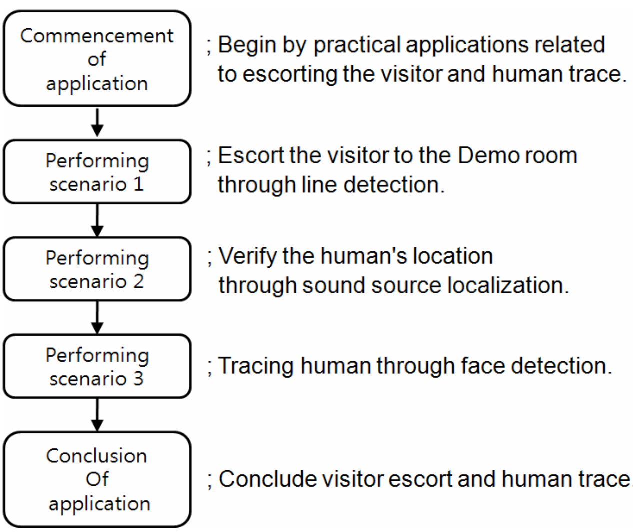


Fig. 7. Flow of application performance

The Demo room to perform scenario 1 is a curvy passage way, and the robot must utilize line detection to guide the visitor to the point of scenario 2. Scenario 2 will automatically begin right after completing Mission 1. Mission completion of scenario 1 is judged when the valid Left/Right line's point of contact value does not transmit to the Slave 2 AVR from line detection FPGA. In scenario 2, the voice calling on the robot by the target of trace is used to detect the direction of the human. Scenario 3 will be automatically performed when sound source localization FPGA transmits output value corresponding to the normal packet to Slave 1. Output from sound source localization by FPGA is used to view the direction of the person standing in scenario 3, and it traces the person via face in the image nearest to the center of the vertical axis. When facial detection FPGA fails in tracing the human, the robot will halt to commence again in scenario 2. The Master AVR of the robot driving part will judge the data transmitted by FPGA through Slaves 1-3 AVR to operate and drive the robot via Slave 0 [Fig. 8].

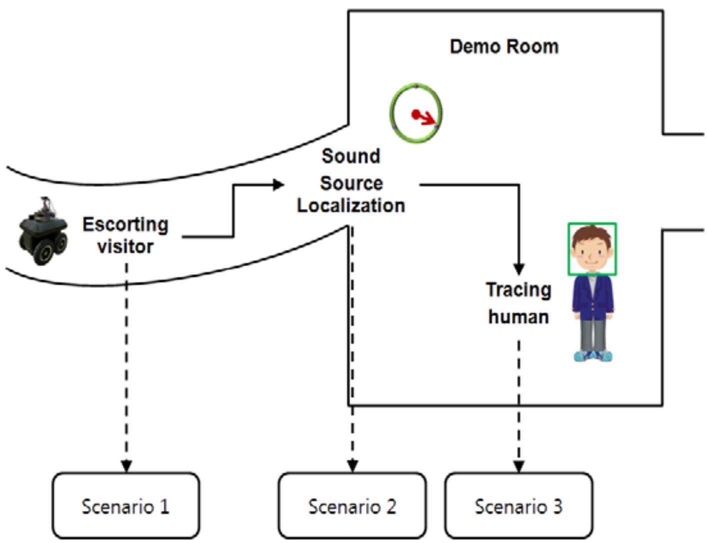


Fig. 8. Scenarios 1-3 performance location

We have successfully performed all scenarios including scenario 1 where navigation on the passage way by line detection using Off-chip based microprocessor and FPGA interface system; scenario 2 confirms the direction of the person through sound source localization when arriving at the Demo room; scenario 3 to continuously trace the human and detect the human face, as is shown in Fig. 9. Below is the photo of a Giant robot equipped with a Camera at the center, with microphone for a sound source localization, and FPGA. The upper right photo is the Microprocessor Unit and the photo in the center is the connection between the personal computer and the Microprocessor Unit. The photo on the left shows a developer framing and monitoring the application on the field. The photo on upper-center to the left is a robot navigating through the passage way by line detection, and the photo right below shows the robot confirming the location of a person via direction detection. The rest of the photos show robots continuously trying to trace the person by face detection.

Fig. 9. Pictures of the end of performance scenario

4 Conclusion

The proposed system, Off-chip based multi-microprocessor and FPGA interface, has successfully performed in all three planned scenarios. Functions realized by the existing FPGA were easily synthesized with a low costing microprocessor in this research, and a speedy robot application framing was achieved. The system proposed in this paper may be applied in various robots in the future. In future work, we plan to develop a system for controlling multiple robots.

Acknowledgments

This research was supported in part by MKE, Korea under ITRC NIPA-2010-(C1090-1021-0008) (NTIS-2010-(1415109527)) and Priority Research Centers Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2010-0020210).

References

1. Li, T.-H.S., Chang, S.-J., Chen, Y.-X.: Implementation of human-like driving skills by autonomous fuzzy behavior control on an FPGA-based car-like mobile robot. IEEE Trans. on Industrial Electronics 50(5), 867–880 (2003)
2. Krohling, R., Zhou, Y., Tyrrell, A.: Evolving FPGA-based robot controller using an evolutionary algorithm. In: 1st International Conference on Artificial Immune Systems, Canterbury (September 2003)
3. Kongmunvattana, A., Chongstivatana, P.: An FPGA-based Behavioral Control System for a Mobile Robot. In: IEEE Asia-Pacific Conference on Circuits and Systems, pp. 759–762 (1998)
4. Bluethmann, W., Ambrose, R.D., Askew, M.S., Huber, E., Goza, M., Rehnmark, F., Lovchik, C., Magruder, D.: Robonaut: A robot designed to work with humans in space. Auton. Robots 14, 179–197 (2003)
5. Hamblen, J.O., Hall, T.S.: System-on-a-Programmable-Chip Technology to Design Embedded Systems. IJCA 13(3), 1–11 (2006)
6. Wolf, W.: The future of Multiprocessor Systems-on-Chips. In: Proc. of Design Automation Conference, pp. 681–685 (2004)
7. Christophersen, H., Pickell, R., Neidhoefer, J., Koller, A., Kannan, S.: A Compact Guidance, Navigation, and Control System for Unmanned Aerial Vehicles. To appear in Journal of Aerospace Computing, Information, and Communication (2006)
8. Panda, P., Dutt, N., Nicolau, A.: On-chip vs. off-chip memory: The data partitioning problem in embedded processor-based systems. ACM Trans. Des. Autom. Electron. Systems 5(3), 682–704 (2001)
9. http://www.ntrex.co.kr/
10. 8-bit AVR Microcontroller with 128K Bytes In-System Programmable Flash datasheet, http://www.atmel.com/
11. http://www.sena.com/
12. Song, T.H., Park, J.H., Jung, S.M., Kwon, K.H., Jeon, J.W.: Embedded Robot Operating Systems for Human-Robot Interaction. In: Asia-Pacific Conference on Computer Human Interaction (APCHI), Incheon, Korea (2008)
13. Jin, S.H., Kim, D.K., Kim, H.S., Lee, C.H., Choi, J.S., Jeon, J.W.: Real-time Sound Source Localization System based on FPGA. In: IEEE International Conference on Industrial Informatics, Daejeon, Korea, July 13-16, pp. 673–677 (2008)
14. Kim, D.K., Jin, S.H., Thuy, N.T., Kim, K.H., Jeon, J.W.: A Real-time Finite Line Detection System Based on FPGA. In: IEEE International Conference on Industrial Informatics, Daejeon, Korea, July 13-16, pp. 655–660 (2008)
15. Jin, S.H., Kim, D.K., Jun, B.J., Kim, D.J., Jeon, J.W.: An FPGA-based Parallel Hardware Architecture for Real-time Face Detection using a Face Certainty Map. In: IEEE International Conference on Application-specific Systems Architectures and Processors (ASAP), Boston MA, USA (2009)

Multi-hop Cooperative Communications Using Multi-relays in IEEE 802.11 WLANs*

Sook-Hyoun Lee, Chang-Yeong Oh, and Tae-Jin Lee**

School of Information and Communication Engineering
Sungkyunkwan University
440-746, Suwon, South Korea, +82-31-290-7982
{sookhyun,ohchy,tjlee}@ece.skku.ac.kr

Abstract. In this paper, we propose a mechanism to increase performance by using cooperative communications in IEEE 802.11. Existing algorithms use one relay between a source and a destination, which is a 2 hop relay. There may be cases that relays cannot be found. We propose a cooperative communication protocol for three hops. The proposed algorithm manages network information (rate among nodes) and selects best relay(s). And it utilizes more than one relay to improve the inefficiency of using one relay by network information. Moreover, relays are given to have the opportunity to send their own data right after relaying the source's data. So relays are compensated for relaying the source's data and the overall throughput is enhanced.

1 Introduction

Multiple-Input Multiple-Output (MIMO) antenna gains spatial diversity [1] while sending and receiving data and improves transmission performance. MIMO may be constrained by size, cost or complexity of mobile devices. Multiple nodes with single antenna can also form a virtual antenna array and it is introduced to mimic MIMO, which is called cooperative communications [2]. Cooperative communications have been proposed to achieve the benefit of spatial diversity.

Cooperative communication protocols to use multiple relays with space time coding [3] and those to use the best relay [4] – [6] have been announced. In the space time-coded protocol, a source transmits data to a destination and cooperative relays overhear and decode it and send it for the source by space time code to the destination. Relays in the repetition-based protocol require their own sub channels for repetition, but relays in the space time-coded protocol utilize the same sub channel, which improves bandwidth efficiency. While performance may degrade as the number of nodes increases in the repetition-based protocol, it is maintained in the space time-coded protocol. It, however, requires more complexity than those schemes with the best relay [7].

* This research was supported by PRCP through NRF of Korea, funded by MEST (2010-0020210), and the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency)(NIPA-2011-(C1090-1111-0005)).

** Corresponding author.

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 120–132, 2011.

In one of the methods using the best relay [4], after a source sends Ready To Send (RTS) and a destination sends Clear To Send (CTS), candidate relays which overhear RTS and CTS calculate the signal strength received from the source and the destination. And the candidate relay nodes start timers based on the channel condition. The timer of the node with the best channel condition is expired first. The node then transmits a short packet to the source and plays the role of the best relay. And in the method called CoopMAC [5], each node collects the rate between itself and neighbor nodes, and saves the rate information in the coopTable. CoopMAC selects an efficient cooperation path by using the coopTable. Nodes with limited resources are required to maintain the table having neighbor information and update the table by exchanging the node states, which is a burden. The aforementioned researches improve performance but may not compensate relays for the use of cooperation resources and for the competition for medium access among relays. In addition, using only one best relay limits the spatial diversity gain due to the constraint on the number of virtual antennas.

In this paper, we propose a new Dual Relay-Cooperative Medium Access Control (DR-CMAC), in which multi-rate (e.g., 1Mbps, 2Mbps, 5.5Mbps and 11Mbps in IEEE 802.11b) nodes are considered. One or two relays are allowed according to the channel states. It uses the rate information among nodes but nodes do not need to keep track of the rate information. We integrate the rate information into the table in Access Point (AP). DR-CMAC can reduce the power the consumption due to removed table maintenance and search. A relay participating in cooperation can also send out its own data while relaying the source data without requiring additional medium access opportunity of the relay. As a result, relay(s) can be compensated for using their own resource. In the legacy IEEE 802.11 [8] without cooperation, low-rate nodes may occupy the medium longer due to fair Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA), which decreases performance. Our DR-CMAC provides a supplementary chance to boost the rate of low-rate nodes with the help of high-rate relays.

The rest of this paper is organized as follows. Section 2 illustrates the proposed algorithm. In Section 3, we analyze the proposed algorithm and conduct simulations. We draw the conclusion in Section 4.

2 DR-CMAC ALGORITHM

The proposed DR-CMAC selects the best path with the minimum transmission time among the direct path, one-hop relay path and two-hop relay path.

2.1 Proposed DR-CMAC Protocol

The proposed DR-CMAC works as follows (see Fig. 1 and Fig. 2).

1. A source node sends a cooperative RTS (CRTS) frame to an AP after back-off time.
2. The nodes with data to the AP overhear the CRTS frame and save the channel state (rate) between the source node and their own. When having an opportunity for transmission, a source node sends the CRTS frame including the rate between the best neighbor node and its own. If the AP receives the CRTS frame from the

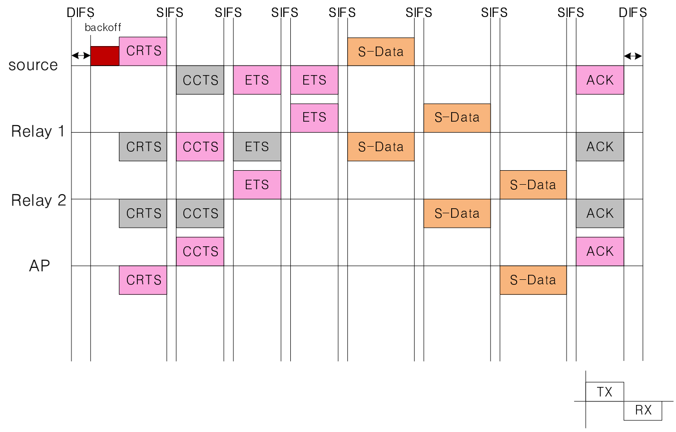


Fig. 1. An example of the proposed DR-CMAC (relay does not have its own data to transmit)

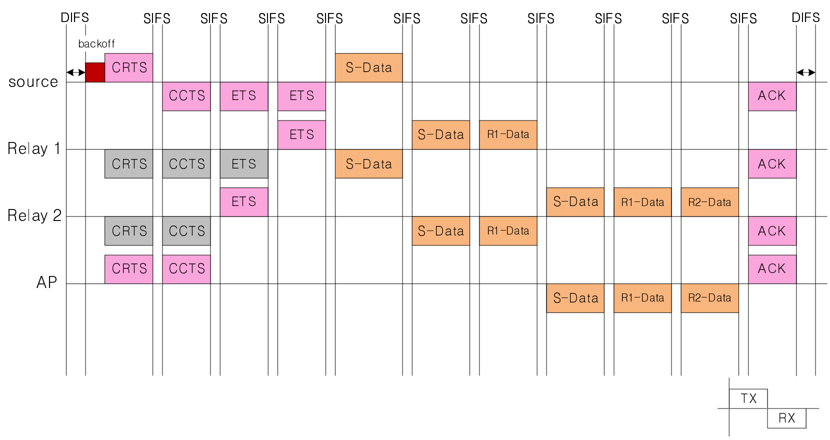


Fig. 2. An example of the proposed DR-CMAC (relay has its own data to transmit)

source, the AP saves the rate between the source and the AP in the MAC header, and the rate information (rate between the source and the best neighbor) in the rate field of the CRTS frame into a table.

3. The AP searches for a path to minimize the transmission time in the table and sends a cooperative CTS (CCTS) frame which includes the relay ID(s) (AID(s)) of the first relay and/or the second relay to the source. We utilize the “more data field” in the MAC header to indicate that one relay or two relays are used.
4. Candidate relays (nodes except the source) overhear the CCTS frame and compare their own AID with the AID in the CCTS frame. If they are matched, the candidate relay(s) become the first relay and/or the second relay.

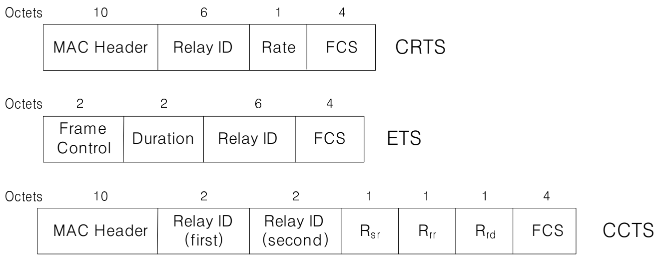


Fig. 3. Proposed frames in our DR-CMAC

5. The first relay and/or the second relay broadcast an Enable to Send (ETS) frame to notify neighbor nodes that they are selected as the relay(s) to support the source with the rate in the CCTS frame.
6. The source, which receives the CCTS frame and the ETS frame, sends data to the first relay and the first relay relays the source's data to the second relay. And the second relay relays it to the AP. In the case of the single relay, the first relay sends the data to the AP directly.
7. If relays have their own data to the AP, they send their own data right after relaying the source's data. If relays do not send their own data to the AP, the AP only sends the Acknowledgement (ACK) frame to the source. Otherwise the AP sends the ACK frame to the source, and the relay(s).

2.2 DR-CMAC's Frame Structure and Relay Path Selection

Frame Structure. We newly define frames for DR-CMAC as in Fig. 3. The CRTS frame acts as the RTS frame with the rate information (rate between the best neighbor node and its own) in the rate field, and the neighbor node's ID in the relay ID field. The nodes selected as relay send the ETS frame to the source. Finally, the CCTS frame works as the CTS frame with the relay information for a source. In our algorithm, the "more data field" in the MAC header is used to classify the single relay or the dual relays. If the more data field is set, the CCTS frame has the information with the dual relays path. Otherwise, the CCTS frame has the single relay path information. The size of the relay ID in the CRTS frame and in the ETS frame is 6 octets, and that in the CCTS frame is 2 octets since the relay ID of the CRTS frame uses MAC address and the relay ID of the ETS frame and CCTS frame uses AID. Nodes send the association request message to an AP and receive their own AIDs from the AP.

Selection of Relay Path. We select a path with the shortest transmission time among the direct path, the relay path with the single relay or the relay path with the dual relays.

T_{direct} is the transmission time from a source to an AP when $8L$ (bits) data is sent by R_{sd} transmission rate.

$$T_{direct} = \frac{8L}{R_{sd}} + T_{PLCP} \tag{1}$$

where T_{PLCP} is the transmission time of the physical header.

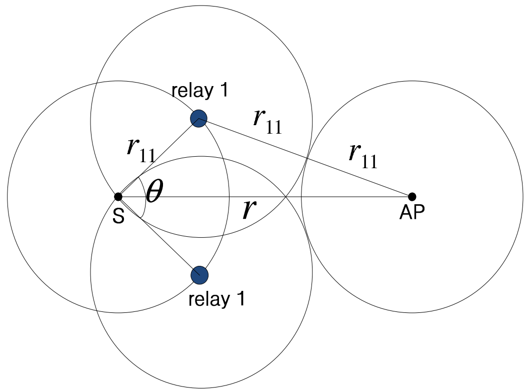


Fig. 4. Available location of relay 1 for 11Mbps transmission from a source to relay 1

T_{single} is the transmission time from a source to an AP with the single relay. The source transmits data to the relay by the rate of R_{sr} and the relay forwards the source's data to the AP by the rate of R_{rd}.

$$T_{single} = \frac{8L}{R_{sr}} + \frac{8L}{R_{rd}} + 2 \times T_{PLCP} + T_{ETS} + 2 \times T_{SIFS} \tag{2}$$

where T_{ETS} is the transmission time of the ETS frame and T_{SIFS} is the time for Short InterFrame Space (SIFS).

T_{dual} is the transmission time from a source to an AP with the dual relays.

$$T_{dual} = \frac{8L}{R_{sr}} + \frac{8L}{R_{rr}} + \frac{8L}{R_{rd}} + 3 \times T_{PLCP} + 2 \times T_{ETS} + 3 \times T_{SIFS} \tag{3}$$

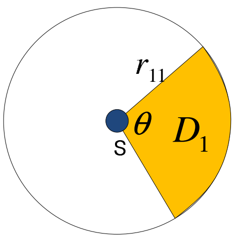
where R_{rr} is the rate from the first relay to the second relay. The source sends data to the first relay and the first relay relays the source's data to the second relay. The second relay relays the source's data to the AP. We select the best path, which has the minimum transmission time among T_{direct}, T_{single} and T_{dual}.

3 Performance Evaluation

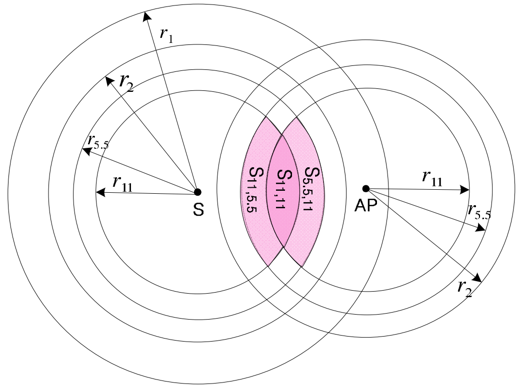
3.1 Location Analysis of Dual Relay

We assume IEEE 802.11b WLANs and analyze the probability that the dual relays are located in a way that the transmission rates of a source to relay 1, relay 1 to relay 2, and relay 2 to AP are 11Mbps, i.e., the highest rate.

Fig. 4 shows the possible positions of the first relay whose transmission rate is 11Mbps from the source to the first relay (relay 1). Since the second relay (relay 2) also has to be able to send data to the AP with the rate of 11Mbps, the relay 1 and relay 2 have to be located within the circle of the 11Mbps transmission rate whose centers are the source and the AP, respectively. And the relay 2 has to be put into the overlapping



(a) Candidate area for the first relay



(b) Candidate area $S_{11,5.5}$, $S_{11,11}$ and $S_{5.5,11}$ for the single relay only

Fig. 5. Location of the first relay

part of the 11Mbps circles whose centers are the relay 1 and the AP. So the relay 1 has to be located within the arc area D_1 as shown in Fig. 5(a). We can find

$$\cos 2 \cdot \frac{\theta}{2} = 2\cos^2\frac{\theta}{2} - 1 = 2\left(\frac{1}{2r_{11}r}(3r_{11}^2 - r^2)\right)^2 - 1. \quad (4)$$

So, θ can be expressed as

$$\theta = \arccos\left\{\frac{1}{2}\left(\frac{1}{r_{11}r}(3r_{11}^2 - r^2)\right)^2 - 1\right\}. \quad (5)$$

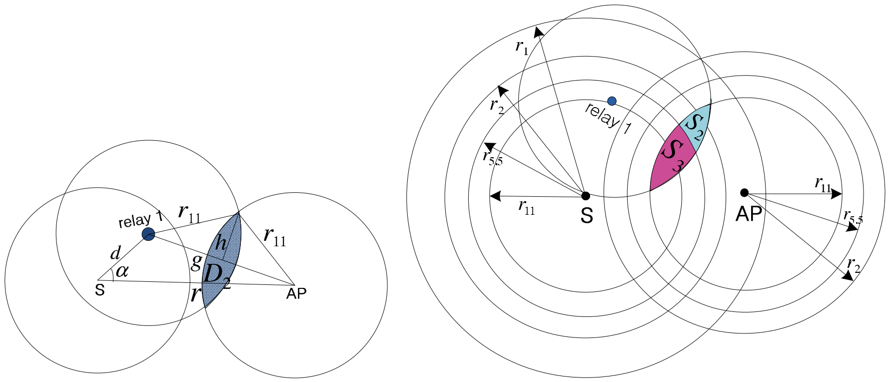
We can express the arc area D_1 as

$$D_1 = \frac{1}{2}r_{11}^2\theta. \quad (6)$$

If the performance with the single relay is better than that with the dual relays, a relay in D_1 will be used as the single relay. The case that the single relay has better performance than the dual relays is when the rate between the source (or AP) and the relay is 5.5Mbps and the rate between the relay and the AP (or source) is 11Mbps, or the rate between the source and the relay is 11Mbps and the rate between the relay and the AP is 11Mbps. We express the cases as $S_{5.5,11}$, $S_{11,5.5}$ and $S_{11,11}$ respectively. Namely, when relay 1 is in the area $S_{11,11}$, $S_{5.5,11}$ or $S_{11,5.5}$, the transmission with the single relay is preferred to the transmission with the dual relays. Therefore, the area D_1 has to exclude $S_{11,11}$ and $S_{11,5.5}$, which is referred to as S_1 (see Fig. 5b). So S_1 would be

$$S_1 = D_1 - S_{11,5.5}(r) - S_{11,11}(r). \quad (7)$$

Fig. 6(a) shows the area in which relay 2 can be located if the relay 1 is located within area S_1, which is referred to as D_2. But, as we have mentioned above, relay 2



(a) Candidate area for the second relay (b) Candidate area S_2 for the second relay

Fig. 6. Location of Second Relay

has to be located on the positions except the area with the single relay. So, the relay 2 can lie in S_2 as in Fig. 6(b), i.e.,

$$S_2 = D_2 - S_3. \tag{8}$$

D_2 can be found as

$$g^2 = d^2 + r^2 - 2dr\cos\alpha, 0 \leq \alpha \leq \frac{\theta}{2}, \tag{9}$$

$$g = \sqrt{d^2 + r^2 - 2dr\cos\alpha}, \tag{10}$$

$$D_2 = 2r_{11}^2 \arcsin\left(\frac{h}{r_{11}}\right) - hg, 0 \leq \alpha \leq \frac{\theta}{2}. \tag{11}$$

We have to find S_3 to evaluate the area S_2. The area S_3 in Fig. 7 is the overlapping part of relay 1's 11Mbps circle, the source's 5.5Mbps circle and the AP's 11Mbps circle.

The points A, B and C are the intersection points of overlapping circles. The x, y coordinates of A, A_x and A_y are [9]

$$\begin{aligned} A_x &= \frac{r_{5.5}^2 - r_{11}^2 + r^2}{2r}, \\ A_y &= \frac{1}{2r}\sqrt{2r^2(r_{5.5}^2 + r_{11}^2) - (r_{5.5}^2 - r_{5.5}^2)^2 - r^4}, \end{aligned} \tag{12}$$

from

$$A_x^2 + A_y^2 = r_{5.5}^2, (A_x - r)^2 + A_y^2 = r_{11}^2.$$

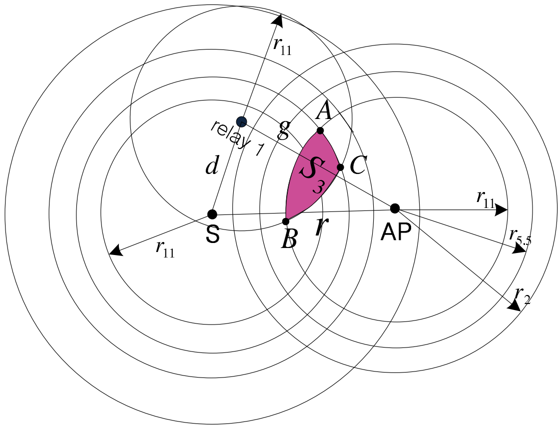


Fig. 7. Area using the single relay only

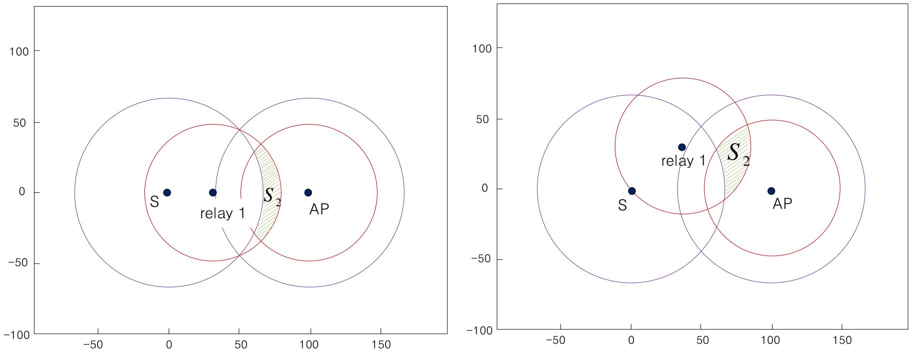
The x, y coordinates of B, B_x and B_y are

$$
\begin{aligned}
B_x &= B'_x \cos\beta - B'_y \sin\beta + r, \\
B_y &= B'_x \sin\beta - B'_y \cos\beta, \\
B'_x &= \frac{r_{11}^2 - r_{11}^2 + g^2}{2g}, \\
B'_y &= \frac{1}{2g}\sqrt{2g^2(r_{11}^2 + r_{11}^2) - (r_{11}^2 - r_{11}^2)^2 - g^4}, \\
\cos\beta &= \frac{r^2 + g^2 - d^2}{2rg}, \sin\beta = \sqrt{1 - \cos^2\beta}.
\end{aligned} \tag{13}
$$

The x, y coordinates of C, C_x and C_y are

$$
\begin{aligned}
C_x &= C'_x \cos\lambda - C'_y \sin\lambda + r, \\
C_y &= C'_x \sin\lambda - C'_y \cos\lambda, \\
C'_x &= \frac{r_{5.5}^2 - r_{11}^2 + d^2}{2d}, \\
C'_y &= \frac{1}{2d}\sqrt{2d^2(r_{5.5}^2 + r_{11}^2) - (r_{5.5}^2 - r_{11}^2)^2 - d^4}, \\
\cos\lambda &= \frac{r^2 + d^2 - g^2}{2rd}, \sin\lambda = \sqrt{1 - \cos^2\lambda}.
\end{aligned} \tag{14}
$$

Then, we find the chord lengths c_1, c_2 and c_3 between A and C, between A and B, and between B and C respectively.



(a) r=100, α=0 (b) r=100, α=0.7

Fig. 8. Candidate area S_2 of relay 2

$$c_1 = \sqrt{(C_x - A_x)^2 + (C_y - A_y)^2}, \tag{15}$$
$$c_2 = \sqrt{(A_x - B_x)^2 + (A_y - B_y)^2},$$
$$c_3 = \sqrt{(B_x - C_x)^2 + (B_y - C_y)^2}.$$

Finally, S_3 is calculated using the chord lengths c_1, c_2 andc_3 as follows:

$$\begin{aligned} S_3 = &\frac{1}{4}\sqrt{(c_1 + c_2 + c_3)(c_3 + c_2 - c_1)} \\ &\cdot \sqrt{(c_1 + c_3 - c_2)(c_1 + c_2 - c_3)} \\ &+ \left(r_{5.5}^2 \arcsin \frac{c_1}{2r_{5.5}} - \frac{c_1}{4}\sqrt{4r_{5.5}^2 - c_1^2} \right) \\ &+ \left(r_{11}^2 \arcsin \frac{c_2}{2r_{11}} - \frac{c_2}{4}\sqrt{4r_{11}^2 - c_2^2} \right) \\ &+ \left(r_{11}^2 \arcsin \frac{c_3}{2r_{11}} - \frac{c_3}{4}\sqrt{4r_{11}^2 - c_3^2} \right). \end{aligned} \tag{16}$$

Thus we find the relay 2's possible area using Eq. (8). Therefore we are able to find the area in which the transmission rates are 11Mbps between the source and relay 1, relay 1 and relay 2, and relay 2 and the AP.

We now calculate the probability with 11Mbps in our DR-CMAC. The area S_2 can change because relay 1 can be put into various locations by changing d and α. Fig. 8 shows two examples. The areas on which relay 1 and relay 2 can be laid are S_1 and S_2, respectively. The relay 2's candidate area S_2 describes the percentage of using dual relays. The probability of dual relays is

$$p_{dual} = \frac{S_2}{\pi r_1^2} \tag{17}$$

In Fig. 8(a), we show S_2 when the distance between the source and the AP is 100m ($r = 100$). And relay 1 is placed on the straight line ($\alpha = 0$) connecting the source and the AP. The distance between the relay and the source is 30m ($d = 30$). And we depict another example in Fig. 8(b) in which the distance between the source and the AP is also 100m, relay 1 is located with the angle of 0.7 ($\alpha = 0.7$) radians and the distance from the source to relay 1 is 48m ($d = 48$). Analysis in Eq. (17) is verified by simulations as shown in Table 1. In the simulation, we measure the cases that the transmission rates are 11Mbps from the source to relay 1, from relay 1 to relay 2, and from relay 2 to the AP when the source and relay 1's location is fixed first and relay 2 is set randomly.

3.2 Throughput of Dual Relay

We calculate throughput when there are dual relays (relay 1 and relay 2) and transmission rates are 11Mbps. We find the transmission time by Eq. (1), Eq. (2) and Eq. (3). But in actual transmission we add average back off time to the transmission time.

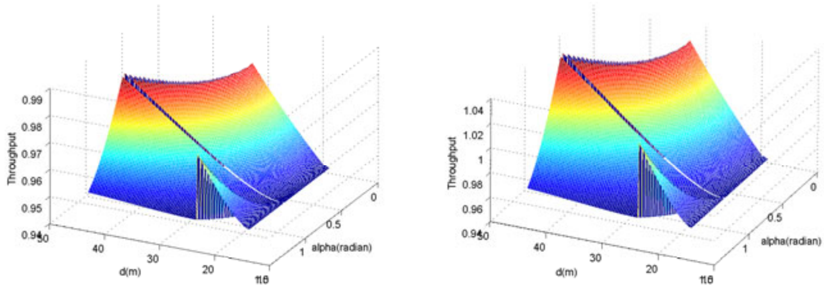
$$\begin{aligned} T_{direct+} &= T_{direct} + T_{slottime} \times E[T_{BACKOFF}], \\ T_{single+} &= T_{single} + T_{slottime} \times E[T_{BACKOFF}], \\ T_{dual+} &= T_{dual} + T_{slottime} \times E[T_{BACKOFF}]. \end{aligned} \tag{18}$$

Table 1. probability(P_{dual}) of 11M-11M-11M transmission

Relay 1's location	Analysis(%)	Simulation(%)
r=100, α=0, d=30	1.97	1.94
r=100, α=0, d=25	0.96	0.95
r=100, α=0.7, d=48	2.29	2.28
r=100, α=0.7, d=40	1.52	1.50

Table 2. Parameters for Simulation

Parameter	Value
MAC header	272 bits
PHY header	192 bits
RTS	352 bits
CRTS	352 + 56 bits
CTS	304 bits
CCTS(single relay, dual relay)	304 + 32 bits, 304 + 56 bits
ETS	272 bits
ACK	304 bits
Rate for MAC/PHY header	1 Mbps
Slot time	20 μs
SIFS	10 μs
DIFS	50 μs
aCWMin	31 slots
aCWMax	1023 slots



(a) Lower bound (b) Upper bound

Fig. 9. Throughput when there are three nodes

We compute throughput for two cases: lower bound and upper bound. The lower bound denotes the case that all relays do not have their own data to transmit to the AP. So relays do not send their own data while acting as relays. Then the transmission data is $8L$ (bits). The lower bound rate

$$R_{lower} = \frac{8 \times L}{T_{dual+}} \times p_{dual} + \frac{8 \times L}{T_{single+}} \times p_{single} + \frac{8 \times L}{T_{direct+}} \times (1 - p_{dual} - p_{single}). \tag{19}$$

We use probability of dual relay and single relay. We show that probability of dual relay and single relay is Eqs. (17) and (20) respectively.

$$p_{single} = \frac{S_{11,11} + S_{5.5,11}}{\pi r_1^2} \tag{20}$$

And the upper bound is the case that all relays have their own data to transmit. So relays send their own data to the AP right after relaying the source's data. So the data to transmit is $3 \times 8L$ (bits) to the AP. And the relay 1 sends the source data and relay 1's data. Lastly, the relay 2 sends the source data, relay 1's data and relay 2's data to the AP. The upper bound rate

$$R_{upper} = 3 \times \frac{8 \times L}{T_{dual+}} \times p_{dual} + 2 \times \frac{8 \times L}{T_{single+}} \times p_{single} + \frac{8 \times L}{T_{direct+}} \times (1 - p_{dual} - p_{single}). \tag{21}$$

We find the throughput with three nodes from Eqs. (19) and (21) in Fig. 9. The distance between a source and an AP is 100m and α ranges from 0 to $\frac{\theta}{2}$ in radians. And the value of d changes from 19m to 48.2m because when $d <$ 19m, the throughput with single relay is better than that with dual relays. So we depict the case that d is greater than 19m. We can see the maximum gap between the lower bound and the upper bound is 5%. We expect the gap becomes larger if there are more nodes.

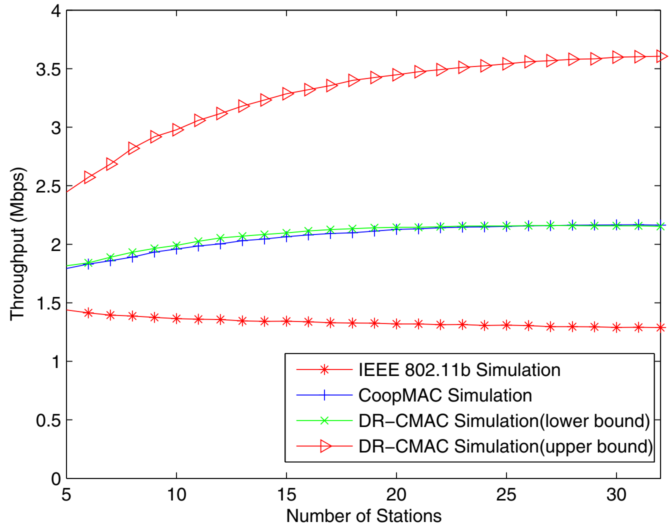


Fig. 10. Throughput of legacy IEEE 802.11b, CoopMAC, and DR-CMAC

3.3 Simulations

We perform simulations for DR-CMAC. Nodes are uniformly distributed around an AP and rates are determined by the distance from the AP. Parameters used in the simulation are shown in Table 2. The distances for the rate 11Mbps, 5.5Mbps, 2Mbps and 1Mbps are set to 48.2m, 61.7m, 74.7m and 100m. Fig. 10 compares the throughput of the legacy IEEE 802.11b, CoopMAC [5] and our DR-CMAC. The throughput of DR-CMAC can range in between the lower bound and the upper bound according to the portion that relays have their own data to the AP. In the legacy IEEE 802.11b, the throughput decreases as the number of nodes increases because the number of nodes with low-rates grows. On the other hand, the throughputs of CoopMAC and DR-CMAC increase. The reason is that the portion of cooperative communications grows as the number of nodes increases. Our DR-CMAC shows performance gain over CoopMAC since relays can send their data without contention while relaying the source's data. If the size of data increases performance gain becomes greater. In our DR-CMAC, low-rate sources utilize fast relaying paths and relays send their own data without medium access competition, which contributes to efficient cooperative communications.

4 Conclusion

In this paper, we have proposed DR-CMAC using cooperative communications in IEEE 802.11. DR-CMAC lessens the spatial diversity restriction by increasing the number of virtual antenna and complements zone, in which single relay or dual relays are allowed. In DR-CMAC, performance increases in low-rate groups 3 and 4 because the nodes receives help from relays. In addition, if there are busy relays having their own data to transmit, performance is improved in all stations because high-rate groups 1 and 2 send their own data without sending further control frames when selected as relay. Therefore we not only accomplish network performance but also relays are compensated for resources because the DR-CMAC gives opportunistic to medium access to high rate nodes which are a few number of nodes comparing with low rate nodes.

References

1. Laneman, J.N., Wornell, G.W.: Exploiting Distributed Spatial Diversity in Wireless Networks. In: Proc. Allerton Conf. Commun., Contr., Computing, Illinois, USA, pp. 1–6 (2000)
2. Nosratinia, A., Hunter, T.E., Hedayat, A.: Cooperative Communication in Wireless Networks. IEEE Commun. Mag. 42(10), 74–80 (2004)
3. Laneman, J.N., Wornell, G.W.: Distributed Space-Time-Coded Protocols for Exploiting Cooperative Diversity in Wireless Network. IEEE Trans. Commun. 49(10), 2415–2425 (2003)
4. Bletsas, A., Shin, H., Win, M.Z.: Simple Cooperative Diversity Method Based on Network. IEEE J. Selected Areas Comm. 24(3), 659–672 (2006)
5. Liu, P., Tao, Z., Narayanan, S., Korakis, T., Panwar, S.S.: CoopMAC: A Cooperative MAC for Wireless LANs. IEEE J. Selected Areas Comm. 25(2), 340–354 (2007)
6. Lin, Z., Erkip, E., Stefanov, A.: Cooperative Regions and Partner Choice in Coded Cooperative Systems. IEEE Trans. Commun. 54(7), 1323–1334 (2006)
7. Hong, Y.-W., Huang, W.-J., Chiu, F.-H., Jay Kuo, C.C.: Cooperative Communications in Resource-Constrained Wireless Networks. IEEE Signal Process. Mag. 24(3), 47–57 (2007)
8. Bianchi, G.: Performance analysis of the IEEE 802.11 Distributed Coordination Function. IEEE J. Selected Areas Comm. 18(3), 535–547 (2000)
9. Fewell, M. P.: Area of Common Overlap of Three Circles, http://www.dsto.defence.gov.au/publications/4815/DSTO-TN-0722.pdf

Adaptive Packing Strategy to Reduce Packing Loss in MF-TDMA Satellite Networks*

MinWoo Lee[1], Jae-Joon Lee[2], Jung-Bin Kim[2], JiNyoung Jang[1], GyooPil Chung[1], and JaeSung Lim[1,2]

[1] Graduate School of Ajou University, SuWon, South Korea
{iminu,powerblow,kp5822,jaslim}@ajou.ac.kr
[2] JangWee Research Institute for National Defence, SuWon, South Korea
{jjnlee,jbkim}@ajou.ac.kr

Abstract. Multi Frequency-Time Division Multiple Access (MF-TDMA) is a multiple access scheme that has widespread use in multimedia satellite networks. A network control center (NCC) receives capacity requests (CRs) from terminals, then it organizes a terminal burst time plan (TBTP) to be distributed efficiently. This plan informs terminals when (timeslots) and where (carrier) to transmit their traffic. Distributing the resource to the terminals is considered to be a bin packing problem. However, sometimes a packing loss may occur when packing the CRs into a TBTP. This means that even if there is sufficient free space, some of the requested cannot be packed. In particular, this occurs more frequently if a total resource is fully requested. If a packing loss occurs, then the terminal whose bit of CRs is lost must transmit the lost part of the initial CRs again. This causes additional time delay for the terminal. This worsens information quality, since the round trip time (RTT) delay is originally large in a satellite network. In this paper, we address the packing loss phenomenon in MF-TDMA for uplinks of satellite networks. We define first packing loss ratio (PLR) and packing loss probability (PLP), then propose an adaptive packing strategy (APS) to reduce packing loss.

Keywords: MF-TDMA, DVB-RCS, Resource Management, Bin Packing Problem.

1 Introduction

Multi frequency-time division multiple access (MF-TDMA) has become a widespread scheme to multiple access in multimedia satellite networks, such as commercial broadcasting, long-range military communication and emergency rescue [1]. MF-TDMA is a suitable selection for a heterogeneous network with various terminals that has different features and antenna size, transmission power, modulation and coding scheme. For example a network with stationary terminals may be assigned bursts that are higher symbol rate, while mobile terminals may be assigned bursts that are of lower symbol rate [2].

* This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency" (NIPA-2011-(C1090-1121-0011)).

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 133–146, 2011.

In this paper, we focus on a packing loss phenomenon that occurs when organizing a terminal burst time plan (TBTP) in MF-TDMA. Packing loss means that even if there is sufficient free space, the requested cannot be fully packed. If a packing loss occurs, then the terminal whose bit of CRs is lost must transmit the lost part of the initial CRs again. This causes additional time delay for the terminal. However, the retransmission is performed after a very long time from the initial capacity request, due to the high delay in satellite networks (i.e. more than 250 msec or 500 msec). When a packing loss occurs, packing efficiency will be worse, since a bit of allocation that is attempted to be packed is lost (i.e. can not be packed). In particular, this occurs more frequently when a total resource is fully requested. Therefore the lost size will increase as the fully loaded duration keeps longer, such as an emergency situation (i.e. a localized warfare, a disaster). This means that many a little loss makes a large loss. [8] is the first work, as far as we know, to describe the packing loss phenomenon and presents several important insights of this. However, the work and related work [4][5][6][7][8] do not present how to reduce this or do not consider the phenomenon. Therefore, we propose an adaptive packing strategy (APS) to reduce the packing loss. APS consists of two algorithms for each side; terminal and NCC. These schemes are easily amenable to recent satellite network systems.

The remainder of this paper is organized as follows. Section 2 describes the background and related work on resource allocation for the MF-TDMA of satellite networks from a packing problem viewpoint. We introduce the packing loss phenomenon, and the necessary steps to reduce packing loss in Section 3. Section 4 presents the adaptive packing strategy to reduce packing loss and its algorithms. Simulation results are presented in Section 5. Finally we conclude this paper in Section 6.

2 Background and Related Work

In Digital Video Broadcasting-Return Channel Satellite (DVB-RCS) standard, Return Channel Satellite Terminals (RCSTs) send message, in which Capacity Request (CR) is contained, to a Network Control Center (NCC) [3]. An RCST may explicitly requests the needed capacity to the NCC in a Demand Assignment Multiple Access (DAMA) scheme. The NCC then allocates return channel timeslots based on each CRs. This is the service as a scheduling and CR evaluation phase as described in Fig. 1. The NCC will allocate the CRs in a two-dimensional resource space as described in Fig. 3 in the resource allocation packing phase. This space is called a Terminal Burst Time Plan (TBTP) that informs to terminals when (timeslots) and where (carrier) to transmit their traffics.

2.1 Related Work for the Packing Problem

Packing the capacity requests (CRs) has been considered as a *bin packing problem*, that is NP-complete. There are some related work about packing resources into the burst time plan of MF-TDMA for a satellite network. Reserve channels with priority (RCP) fit [4] is considered in a fixed bandwidth size. In [5], various packing algorithms are considered such as first fit, best fit, worst Fit and next Fit for DVB-RCS. Some hybrid bin-packing algorithm are proposed, which have focused on improvement of packing

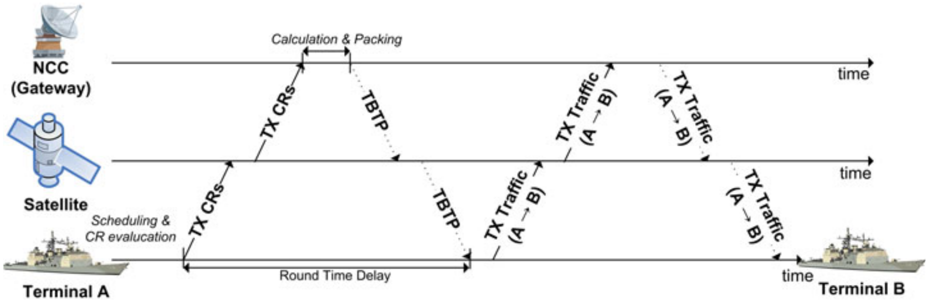


Fig. 1. Overall CR and Packing Procedure

efficiency and QoS required by networks such as delay and jitter fulfillment [6], [7]. They even don't consider the packing loss phenomenon. But in [8], the author shows the probability of not fully packing and three important insights about this as follows. First, when all terminals are using same carrier size, capacity requests are packed perfectly at all resource utilizations. Second, there is a trend carrier sizes that more terminals using small carrier size less packing loss occurrence. Third, the resources are packed completely up to a resource utilization of 80 % of the system. Beyond this point, compositions with terminals using larger carrier size often do not pack completely. This means that a novel scheme should be needed to mitigate the incomplete packing trend.

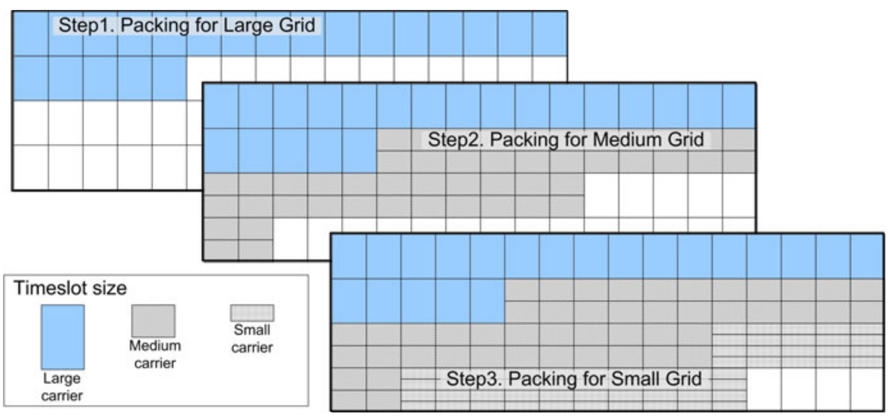


Fig. 2. Example of Grid Packing with three carrier sizes (*Large, Medium, Small*)

2.2 Grid Packing Algorithm for MF-TDMA

The bin packing problem for the resource allocation can be approached as an optimization problem in which each terminal wants to maximize its capacity utilization. However, finding the highest value solution may require a large number of calculations, in general it may not be computationally feasible for systems to find the optimal solution. A simple and robust packing algorithm is proposed in [8] using grid methods that we consider in this paper.

In Fig. 2, there are three grids for each of three carrier sizes. Each grid has two-dimension blocks; the row is for timeslots and the column is for carriers. According to the carrier size, the grid has to be used differently. All the grids have a different number of rows corresponding to W_{SF}/W_j, where W_{SF} is the total bandwidth of a superframe and W_j is the required bandwidth (carrier size) for a terminal j. The packing algorithm can be summarized as below.

1. Select a grid : Select the grid G_j corresponding to the bandwidth requested from CRs. (All CRs are sorted by the decreasing order of request size in each 'request table').
2. Take the request table : The highest priority (i.e. CR size or other weight value) CR that does not conflict with a current allocation to this terminal is taken off the request table for this grid G_j.
3. Adjust the other grid's request table : All other request table of all other grids are taken off also to remove the corresponding allocation from those request tables.

3 Packing Loss Phenomenon

In this paper, we assume that each terminal has only one modulator. This means that only one carrier at a time can be used, since it is an important issue to minimize the cost of the terminals [5]. Packing loss differs from the blocking that occurs due to the lack of capacity. The former occurs due to the restrictions in a communication system, not due to the capacity. Packing loss means that even if there is sufficient free space, the requested cannot be fully packed. Packing loss may occur, since any satellite communication system has constraints on the packing. In this paper, we consider the avoidance of parallel timeslot assignment, that is considered commonly in the above related work [4][5][6][7][8]. Fig. 3 shows an example of parallel timeslot assignment that is not allowed here.

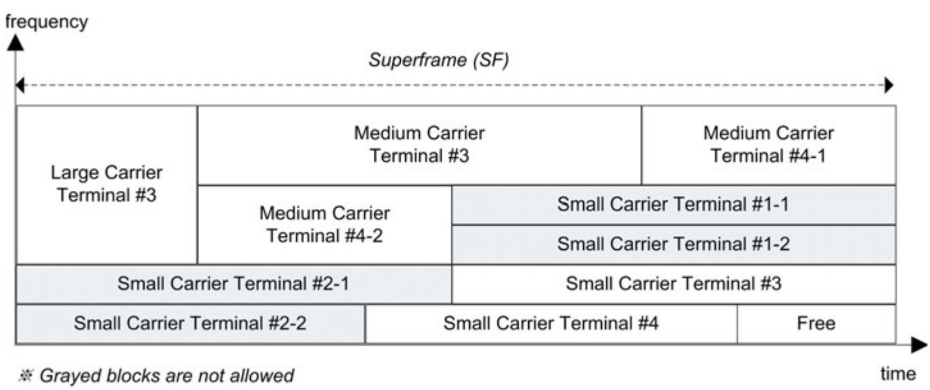


Fig. 3. Example of parallel timeslot assignments

Due to the restriction, an actual resource utilization may differ from an attempted resource utilization. These are defined as follows.

$$\text{Attempted Resource Utilization } (\rho') = \frac{\text{total capacity requests (CRs)}}{\text{total capacity of TBTP}} \tag{1}$$

$$\text{Actual Resource Utilization } (\rho) = \frac{\text{packed capacity requests (CRs)}}{\text{total capacity of TBTP}} \tag{2}$$

We also define packing loss ratio (PLR) and packing loss probability (PLP). The former means the ratio between an attempted resource utilization (ρ') and an actual resource utilization (ρ). The latter means the probability of the occurrence of the packing loss. PLR in ith TBTP can be defined by the equation (3). When packing loss occurs, PLR_i is more than 0, otherwise PLR_i is 0.

$$\text{PLR}_i = 1 - \frac{\text{Actual Resource Utilization}(\rho)}{\text{Attempted Resource Utilization}(\rho')} \tag{3}$$

PLP in ith TBTP can be presented also as follows. If we let L_i and $I_i(k)$;

$$L_i = \begin{cases} 1 & \text{if PLR}_i > 0 \text{ in } i\text{th TBTP,} \\ 0 & \text{otherwise,} \end{cases}$$

$$I_i(k) = \begin{cases} 1 & \text{if } \rho'_i = k \text{ in } i\text{th TBTP,} \\ 0 & \text{otherwise.} \end{cases}$$

then $\sum_{i=1}^{n} L_i$ represents the number of packing loss occurrence in n TBTPs, and $\sum_{i=1}^{n} I_i(k)$ is the number of the attempted resource utilization when is $\rho' = k$, respectively.

Hence;

$$\text{PLP} = \frac{1}{n} \sum_{i=1}^{n} L_i \tag{4}$$

$$\text{PLP}(k) = \frac{\sum_{i=1}^{n} L_i \times I_i(k)}{\sum_{i=1}^{n} I_i(k)} \tag{5}$$

4 Adaptive Packing Strategy (APS)

4.1 Adjusting Carrier Size to Reduce Packing Loss

We propose to adjust carrier size (bandwidth) in a terminal to reduce packing loss. This is like rotating the demand rectangles 90° in the Two Dimensional Cutting Stock Problem (2DCSP), as shown in Fig. 4. A capability to adjust carrier size is a similar concept with the Fade Mitigation Techniques (FMT). To keep the symbol rate constant under the use of a more robust Forward Error Correcting (FEC) code rate that results in a reduction of the information data rate, it is required to expense of a larger bandwidth [9]. This means a terminal is capable of adjusting own carrier size.

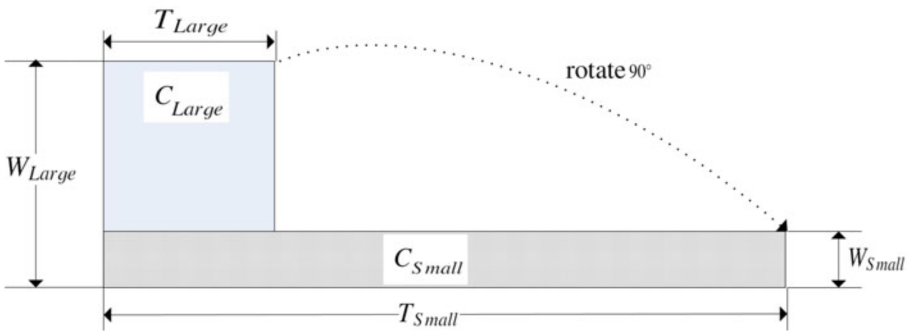


Fig. 4. The relationship of symbol rate and bandwidth (carrier size)

Now, we present the condition to use APS. For each terminal, if terminal j requests a bandwidth (W_j) with transmission power (P_j) at a superframe, then the capacity of terminal j is given by;

$$C_j = W_j \log\left(1 + \frac{P_j}{N_0 W_j}\right) \qquad [\text{sps}] \tag{6}$$

Then, we can calculate the number of timeslots (N_T) to adjust carrier size of terminal j from large to small as below, where C_{Large} is a capacity when terminal j uses a large carrier size and C_{Small} is a capacity when the terminal uses a small one, and T_{SF} is the total timeslots for small carrier size in a superframe or a TBTP.

$$\left\lceil \frac{C_{Large}}{C_{Small}} \right\rceil \times N_T \leq T_{SF} \tag{7}$$

4.2 APS for Terminal j Side

APS for a terminal j side is described using a flowchart in Fig.5. Capacity requests generated in terminal j are sent using the Satellite Access Control (SAC) messages defined in [3]. If terminal j is allowed to use various bandwidth (carrier size), then it will be a candidate terminal. This is indicated in the SAC message. We can use the 4 bits in the field *Channel_ID* at the SAC messages, as in [10], so that let NCC know if terminal j is a candidate terminal and it is able to use another carrier size(s).

When a terminal receives a TBTP from NCC then it has to check if own allocated carrier size is identical with the requested before. If the allocated carrier size has been changed with one of candidate carrier size(s) in own list, then it has to adjust its carrier size. Before the terminal transmit its traffic, it has to calculate the number of timeslots again, as the Eq. (7).

4.3 APS for NCC Side

APS for an NCC side is described using the flowchart in Fig. 6. CRs from all terminals are aggregated at NCC for a TBTP. The request table will be updated and sorted in

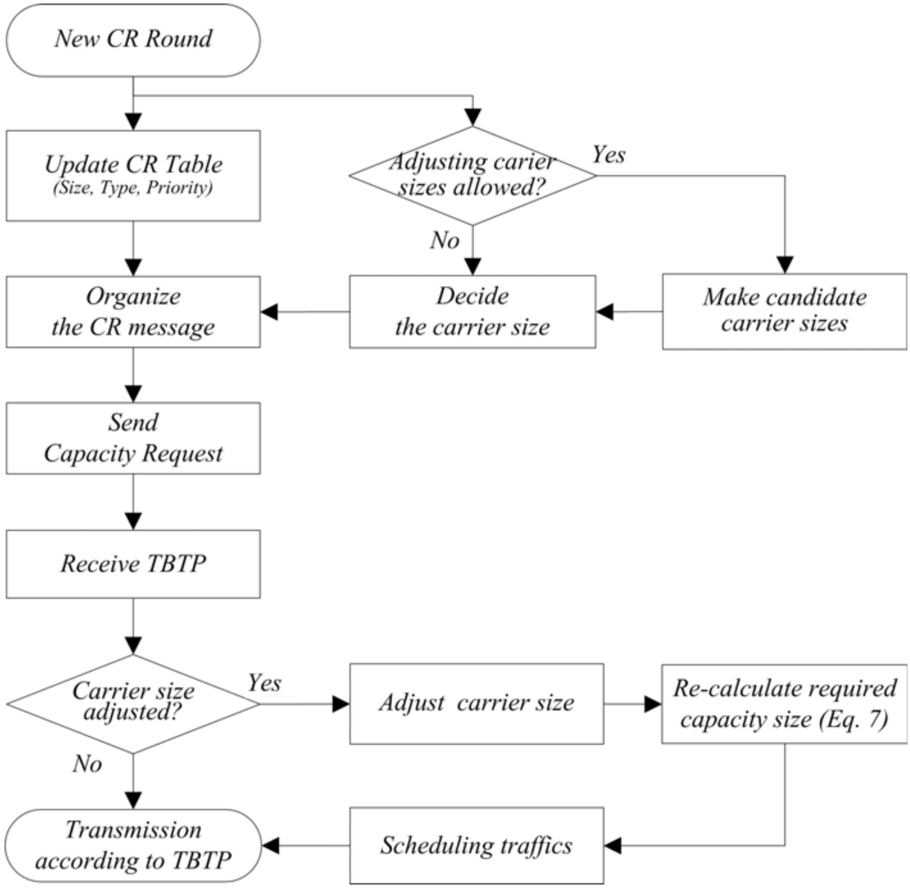


Fig. 5. Adaptive Packing Strategy Procedure for Terminal Side

decreasing order of CR size. Then, NCC can calculate the attempted resource utilization (ρ') that is the aggregated CRs to the total capacity of a superframe in this TBTP.

Simultaneously, the NCC can make a list of candidate terminals that are able to adjust their carrier size(s) from the SAC messages. The NCC can know if packing loss occurs by comparing the actual resource utilization (ρ) and the attempted resource utilization (ρ'). If packing loss occurs, then the NCC checks the candidate terminal list and finds a suitable candidate terminal by which it is able to reduce packing loss. A suitable candidate terminal is a terminal that can use a smaller carrier size.

The suitable candidate terminal is decided by the two steps as follows. First, NCC searches for a candidate terminal from the candidate terminal list in which candidate terminals are sorted by carrier size in decreasing order. Second, if there are more than two candidate terminals using same carrier size, then NCC select a candidate terminal whose request is the smallest among them. This will minimize the number of requested timeslots after the recalculation. If the NCC finds a suitable candidate terminal, then it has to modify the TBTP again with the candidate carrier size. If the packing loss is

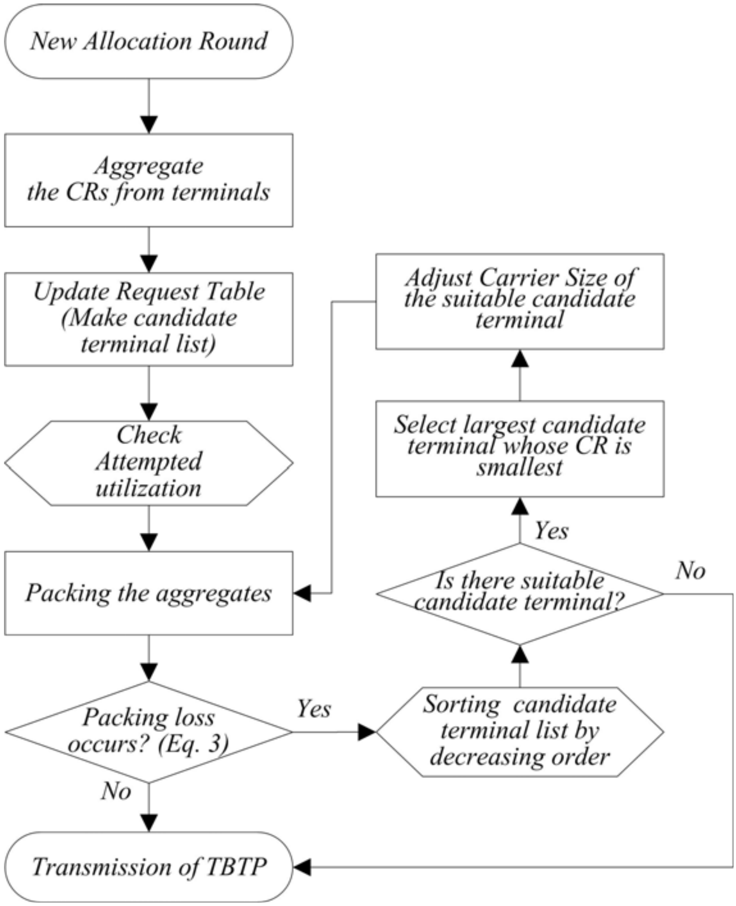


Fig. 6. Adaptive Packing Strategy Procedure for NCC Side

reduced, then the NCC transmits the modified TBTP. Otherwise, even though packing loss occurs, the NCC transmits the original TBTP. The APS is so effective that the latter case may not be problem. Since our experiments in Section 5 show that when there is only one candidate terminal, packing loss will be reduced.

4.4 A Simple Packing Scenario

We give a simple packing scenario to illustrate how APS can reduce packing loss and increase actual resource utilization. Assume that CRs come from five terminals such as $A_{(4,3)}$, $B_{(2,12)}$, $C_{(2,5)}$, $D_{(1,10)}$ and $E_{(1,8)}$ as shown in Fig. 7. In this case, $A_{(4,3)}$ means that terminal A requests a capacity with three timeslots for a large carrier size (i.e. $W_A = 4$). Total resources are 64 timeslots. Attempted resource utilization (ρ') is 1.00, however, actual resource utilization (ρ) is 0.98 in this case. Because one of the requests from terminal D is lost, as shown in Fig. 7. If terminal A is a candidate one that can adjust

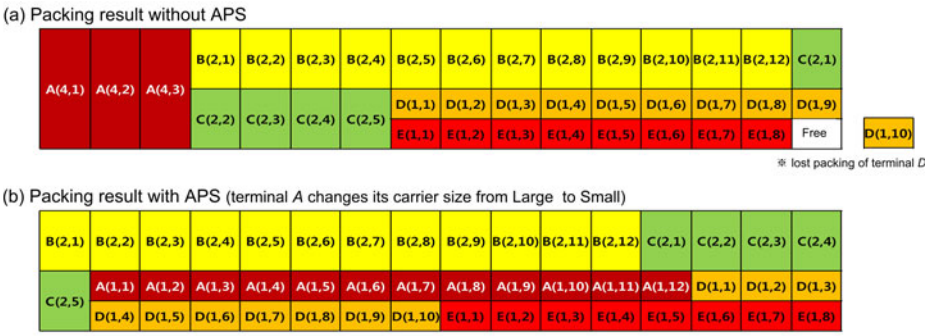


Fig. 7. Simple APS example

its carrier size with $W_A = \{4, 1\}$, then the actual resource utilization is increased to 1.00 and PLR reduced from 0.02 to 0.00 (i.e. no packing loss).

5 Simulation Results

In this section, we present two experiments that are primarily concerned with the occurrence of packing loss and the effectiveness of the APS. The system we consider here is based on the military satellite communication (MILSATCOM) that is described in [8]. In this system, the total resource for a superframe (or a TBTP) is composed of bandwidth W_{SF} (= $48 \times W_{Small}$) and time T_{SF} (= 72 timeslots). The system has three types of carrier sizes such as *Large* (W_{Large}), *Medium* (W_{Medium}) and *Small* (W_{Small}). Thus, all of the terminals share 3,456 timeslots ($W_{SF} \times T_{SF}$=48×72) in the small carrier aspect. These are related to each other as below;

$$W_{Large} = 12 \times W_{Small}, \qquad W_{Medium} = 3 \times W_{Small} \tag{8}$$

5.1 Occurrence of Packing Loss

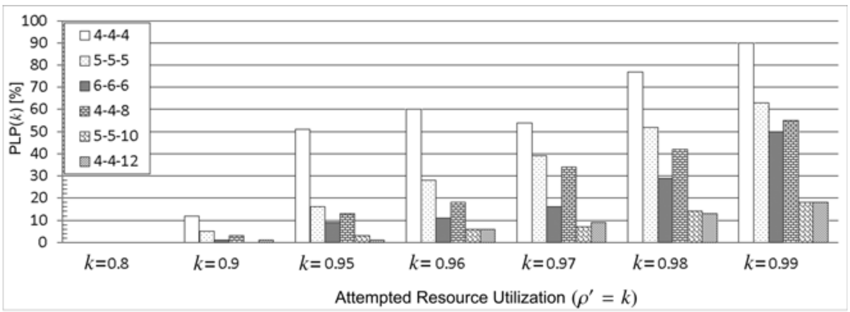
We experimented on packing loss with a broad range of requested timeslots first. The terminal sets are composed of three carrier sizes such as defined above with different terminal numbers. Each terminal generates the number of required timeslots (i.e. CRs). In this simulation, the number of CR is generated randomly drawn from the standard normal distribution between 0 to 72 a broad range to show the occurrence of packing loss for various situation. We ran the simulation 5,000 times for each set to inspect the packing loss phenomenon with various terminal compositions. We made eight ratios from 1:1:1 to 3:3:1 by changing terminal numbers.

We present here the typical packing results in table 1. This shows some typical packing results whose PLP are highly ranked among the experiments. The results show that it is difficult to predict the PLP of any terminal set, since there are too many factors related with each other; such as ratio of carrier sizes, terminal numbers and requested timeslot numbers. An average PLR is the ratio of the lost timeslots to the total timeslots in the small timeslots aspect (i.e. 3,456 timeslots). In general cases, any small Avg.

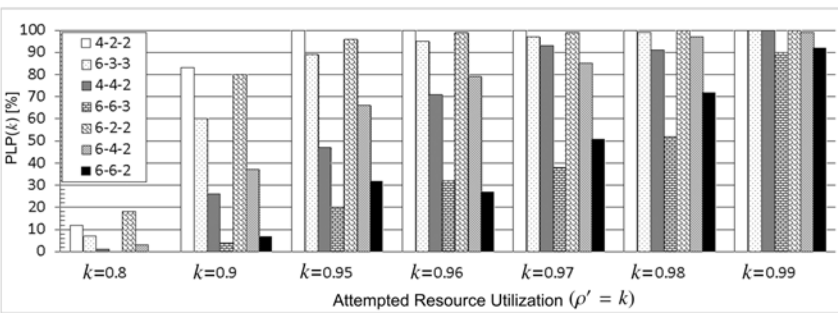
Table 1. Typical packing results with broad range of requested timeslots

Ratio of Sets	Large	Medium	Small	PLP	Avg. PLR	Max. PLR
1:1:1	4	4	4	5.68%	0.57%	4.00%
	5	5	5	14.10%	0.40%	3.00%
1:1:2	4	4	8	4.26%	0.31%	2.00%
	5	5	10	3.04%	0.15%	1.00%
1:1:3	4	4	12	2.38%	0.12%	1.00%
2:1:1	4	2	2	4.66%	1.02%	5.00%
	6	3	3	43.02%	1.08%	4.00%
2:2:1	4	4	2	5.90%	0.78%	3.00%
	6	6	3	10.06%	0.42%	3.00%
3:1:1	6	2	2	50.42%	1.63%	5.00%
3:2:1	6	4	2	39.22%	0.80%	4.00%
3:3:1	6	6	2	14.40%	0.51%	3.00%

[N.B.] Avg.(Max.) PLR : Average(Maximum) of Packing Loss Ratio.



(a) Ratio of CRs (1:1:1 ~ 1:1:3)



(b) Ratio of CRs (2:1:1 ~ 3:3:1)

Fig. 8. PLP distribution over Attempted Resource Utilization (ρ')

PLR may not be problem if the PLP also is small. However, if even a small PLR occurs frequently, then the sum of the PLR will increase for the total duration. This can be verified from the Fig. 8, that shows the distribution of PLP(k) over the attempted resource utilization ($\rho' = k$). In our experiment and the MILSATCOM environment of

[8], the resources are packed completely up to $\rho' = 0.80$. And most of the packing loss occurs, when the resource utilization exceeds $\rho' = 0.80$ of the total capacity. This trend is identical with the results of [8].

As mentioned before, if any emergency event occurs, such as a localized warfare or a disaster, then the attempted resource utilization will increase rapidly. Above all things, since it is of high interest to maximize the utilization for the satellite network resource, it is assumed that a TBTP is fully packed as much as possible. Therefore, if these fully loaded events persist for a long time, then packing loss will occur frequently (i.e. high PLP) and the summation of the lost numbers will increase. This is the reason that we propose the adaptive packing strategy (APS) to reduce the packing loss.

5.2 Effectiveness of APS

In this second experiment, we use the tighter ranges of requested timeslots as described in table 2, since the terminal numbers and the ranges of CRs should be more practical. The number of terminals is limited to 16 per carrier size. The range of requested timeslots is dependent on the allocated carrier size, because a candidate terminal should use the re-calculated timeslots to adjust its carrier size according to the the Eq. (7). Therefore, the requested timeslots are limited with the specified mean of capacity requests and tight variance to be fully loaded of the worst cases. In fact, the set (16-16-16) is an appropriate composition to be able to generate CRs of average 94.2% of the attempted resource utilization.

We set up four cases with one non-APS case (16-16-16) and three APS-applied cases. Each case is composed of 48 terminals with three types of carrier size. The sequence of carrier sizes is such as *Large-Medium-Small*. In this simulation, we define two indications. '16(1)' means one of 16 terminals is a candidate one that is able to adjust carrier size, such as *Large* → *Small* or *Medium* → *Small*. '16(*)' means that the original (non-APS) carrier size will be adjusted to this carrier size, according to the APS. For example, 16(1)-16-16(*) means that one of the 16 *Large* terminals will be adjusted to a *Small* one. 16(1)-16(1)-16(*) means that one of the 16 *Large* and one of the 16 *Medium* terminals will be adjusted to two *Small* terminals.

Fig. 9 shows the number of packing loss occurrence over the attempted resource utilizations ($0.959 \leq \rho' \leq 0.977$). The total number of capacity requests from the 48 terminals is 5,000 times at each experiment. This shows that a PLP(k) is related with the attempted resource utilization and it will have a high value in a fully loaded environment. However, two APS-applied sets, 16(1)-16-16(*) and 16-16(1)-16(*), show no changes or reduced PLP(k) at most of the ranges.

In Fig. 10 (a), (b), (c) and (d), the effectiveness of the APS is shown. According to Fig. 10 (a), the non-APS set (16-16-16) shows packing loss with PLP=3.54 [%].

Table 2. The specified traffic distribution

Type	Carrier size	Mean of CR	Var. of CR	Num. of Terminals
Large	12 x Small	5.1	0.75	16
Medium	3 x Small	23.2	0.75	16
Small	1 x Small	71.6	0.55	16

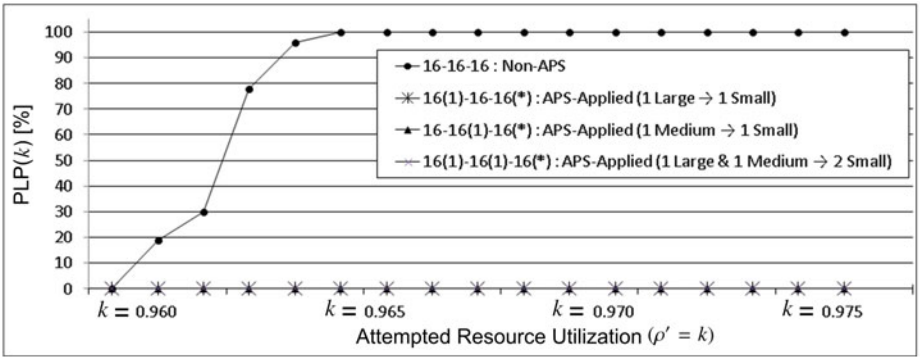


Fig. 9. Experimental simulation results comparing the PLP(k)s with APS-applied sets over Attempted Resource Utilization (ρ') (Total TBTP Numbers : 5,000).

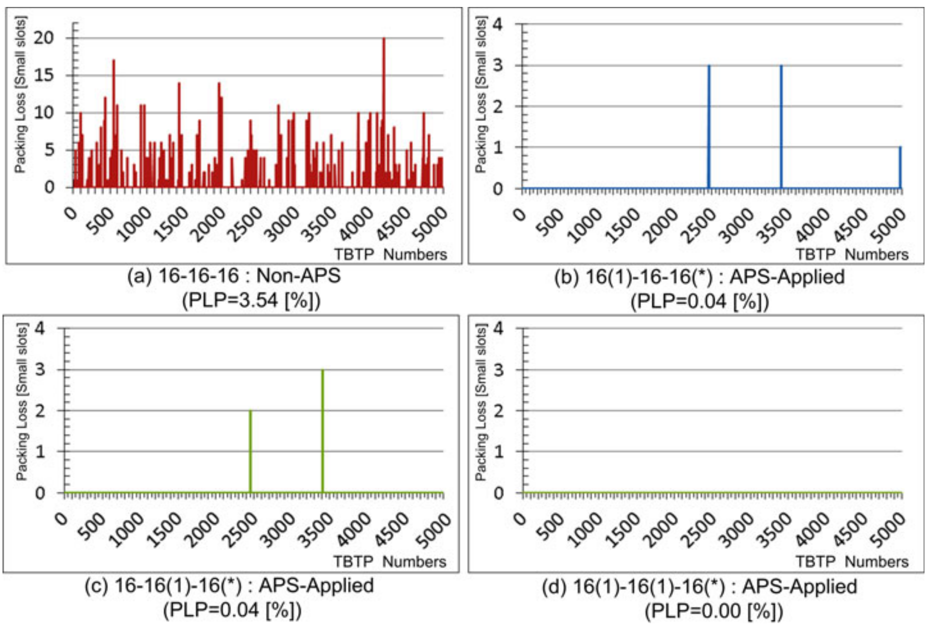


Fig. 10. Overall PLPs with APS-applied sets

In Fig. 10 (b) and (c) for APS-applied sets, 16(1)-16-16(*) and 16-16(1)-16(*), PLPs are reduced (i.e. 0.04 [%]) overall the range of TBTP numbers. Moreover as shown in Fig. 10 (d) we have also experimented the efficiency of a third special case with 16(1)-16(1)-16(*) set. In this case we have adjusted two types of carrier sizes *Large* and *Medium* simultaneously manner, which shows better results than adjusting only single carrier size. The APS can also support to adjust more than two carrier types, since the APS algorithm can be iterated at an NCC.

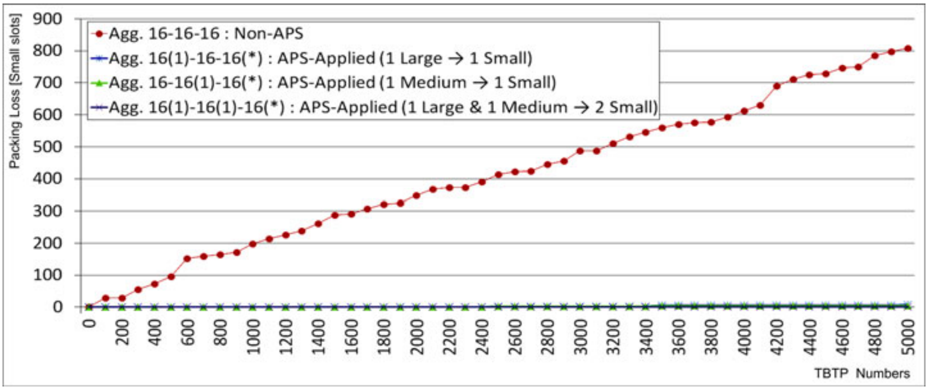


Fig. 11. Aggregated Quantities of the Packing Losses

Finally, Fig. 11 shows the aggregated quantities of the packing losses. The aggregated quantities for non-APS are steadily increasing with respect to other APS-applied cases. Therefore, it can be easily seen that the APS efficiently maintains the aggregated packing losses as compared to increase in the TBTP numbers.

6 Conclusions

In this paper, we focused on the packing loss phenomenon that occurs when an NCC organizes a Burst Time Plan (BTP) for MF-TDMA in satellite networks. Packing loss has not been studied in-depth, since a fixed MF-TDMA structure is mostly used. However, heterogeneous traffic and rain attenuation should be considered more, then satellite networks should operate with various carrier sizes and a dynamic MF-TDMA structure. Therefore, the packing loss phenomenon easily occurs frequently, as shown in our simulation results. We hope the network administrators will consider the packing loss when they plan the composition of terminals and traffic model. We propose an adaptive packing strategy (APS) to reduce the packing loss. And we show the condition to use APS with adjusting carrier size. The experimental simulation results show that APS reduce packing loss probability (PLP) and packing loss ratio (PLR) effectively.

References

1. Kota, S.L.: Digital Video Broadcasting(DVB) Networks: Architectures, Emerging Standards, and Applications. In: MILCOM (2009)
2. Wiss, J., Gupta, R.: The WIN-T MF-TDMA mesh network centric waveform. In: MILCOM (2007)
3. ETSI EN 301 790 V1.5.1 (2009-05)
4. Park, J.M., Savagaonkar, U., Chong, E.K.P., Siegel, H.J., Jones, S.D.: Allocation of QoS connections in MF-TDMA satellite systems: A two-phase approach. IEEE Transactions on Vehicular Tech. (2005)

5. Kissling, C.: Efficient Resource Management for a GEO MF-TDMA Return Link with Channel Adaptation. In: 24th AIAA International Communications Satellite Systems Conference, ICSSC (2006)
6. Petraki, D.K., Anastasopoulos, M.P., Cottis, P.G.: Dynamic resource allocation for DVB-RCS networks. International Journal of Satellite Communications and Networking 26(3), 189–210 (2008)
7. Yao, H., Royster IV, T., McLamb, J., Mustafa, M., Yazdani, N.: Jitter-Aware Time-Frequency Resource Allocation and Packing Algorithm. In: MILCOM (2009)
8. Yazdani, N.: Multi-Frequency Time-Division Multiple-Access (MF-TDMA) Resource Packing. In: MILCOM (2008)
9. Petraki, D.K., Anastasopoulos, M.P., Panagopoulos, A.D., Cottis, P.G.: Dynamic resource calculation algorithm in MF-TDMA satellite networks. In: 16th IST Mobile and Wireless Communications Summit (2007)
10. Morell, A., Seco-Granados, G., Vazquez-Castro, M.A.: Cross-Layer Design of Dynamic Bandwidth Allocation in DVB-RCS. IEEE Systems Journal (2008)

MyUT: Design and Implementation of Efficient User-Level Thread Management for Improving Cache Utilization

Inhyuk Kim, Eunhwan Shin, Junghan Kim, and Young Ik Eom

School of Information and Communication Eng., Sungkyunkwan University,
300 Cheoncheon-dong, Jangan-gu, Suwon, Gyeonggi-do 440-746, Korea
{kkojiband,comsky,gtgkjh,yieom}@ece.skku.ac.kr

Abstract. The appearance of the multicore processors and the advancement of multithread programming have lead the new paradigm of the system optimization. Especially, the cache performance has been known as the one of the most important factor of the system optimization. The user-level thread management, the unvirtualized thread and the exception-less system call are introduced to improve the cache utilization of the multithread programming and parallel programming. However, these schemes have some limitations on applications domain. So, we propose the general purpose user-level thread management scheme to reduce the context-switch, CPU-migration and synchronous system call which pollute the amount of caches. We show evaluation of our system on the three workloads. We show the performance improvements of about 10-20% in respect of the CPU, memory and IO intensive workloads and analyze the effects of the three policies and techniques through the experiments.

Keywords: User-level thread, Unvirtualized thread, Exception-less system call, Cache utilization.

1 Introduction

The multicore processors have been spread recently not only desktop computers but also mobile devices, and many research has been studied for making the best use of the multicore processors. Especially, the efficient thread management is the one of the most important subjects, since it is used for multithread program and parallel programming basically. The kernel-level thread which is widely used usually is heavy and the kernel-level context-switch is expensive, so the user-level thread management schemes have been introduced up to recently. However, the outcomes of these research have some limitations on applications domain [1][2].

The cache utilization is very important factor in modern processors. The more cache hits, the more performance improvements. However, there are many harmful behavior to the cache utilization such as the context-switch, CPU-migration and synchronous system call. So, to reduce these behavior guarantees the more efficient cache utilization [3].

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 147–156, 2011.

Our goal is to develop the general purpose user-level thread management scheme for efficient cache utilization. To achieve this goal, we designed and implemented new user-level thread management scheme, MyUT to reduce unnecessary contentions between the user-level threads using the **no-contention scheduling policy**. And we applied the **unvirtualized thread** and the **exception-less system call** to the MyUT. So, the MyUT improved the cache utilization by reducing the context-switch, CPU-migration and synchronous system call. And we evaluated the three workloads and environments using the MyUT to verify whether the MyUT is proper for the general purpose thread management scheme.

This paper includes following. The Section 1 shows the factors which affect the cache utilization through the experiments. The Section 2 contains the related works and the Section 3 describes the proposed scheme, MyUT in detail. The Section 4 presents the results of three experiments and the Section 5 concludes the paper.

2 The Sources of Cache Misses

In modern computer architecture, the cache utilization is one of the most important for performance of the systems. Especially, the multicore processors have recently begun to spread to desktop computers and mobile devices, so the research and development of the cache optimization schemes have been studied steadily until now. Before describing the cache optimization schemes, we will introduce the several problems to decrease the utilization of caches in detail.

2.1 Cache Coloring

The cache coloring of virtual memory subsystem is the typical problem to increase cache-misses ratio and performance non-determinism which places an important constraint on the accuracy of benchmark. Because the translation of virtual address to physical address by paging decreases spatial locality of caches. There are several schemes to resolve the cache coloring which are employed in operating systems such as Solaris, FreeBSD and WindowsNT. However, several problems are still remained such as recoloring and page cache coloring [4].

2.2 Cache Pollution

The cache pollutions are occurred by basic behaviors of operating systems such as context-switch, CPU-migration, system call and interrupt handling. These behaviors caused the contentions between processes, and between process and kernel on the caches such as the L1 data/instruction caches, the L2/L3 unified caches, branch prediction tables and translation look-aside buffers. We had some experiments to evaluate how much pollutions are occurred by these basic behaviors [5].

The first experiment is to evaluate how much harmful CPU-migration can be on cache/branch misses. We got the result of the case with no CPU-migration using *sched_setaffinity* function with the argument of every time same CPU number and the other case with CPU-migration using *sched_setaffinity* function with the argument of every time different CPU number for eliminating the cost of system call. The next experiment shows the effects of system call on cache/branch misses. We compared no system call, **null** and **write** system calls for obtaining the direct cost and indirect cost of system call. The last experiment is performed for the influence on the processor caches of external interrupt handling. For this, we used APIC(Advanced Programmible Interrupt Controller) for controlling the delivery of the IRQ requests to the processors. The results of experiments presented in the Figure 1 [6][7].

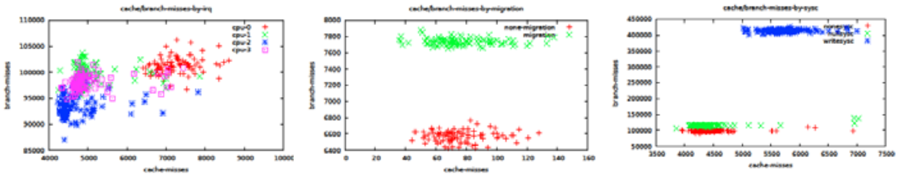


Fig. 1. Cache/Branch misses by CPU migration, system call and interrupt handling

Above three graphs show the distributions of the cost of the basic behaviors of operating systems. The CPU-migration doesn't affect the cache misses because of L2/L3 unified caches, but it affects the branch prediction tables related with processor pipelining. The difference between no system call and **null** system call is the direct cost of system call, and the indirect cost of write system call which is the difference between **null** and **write** system call shows by how much cache/branch misses are occurred on the system call handler. The last graph shows the CPU-0 which handles almost the IRQ requests has more cache/branch misses than other [8].

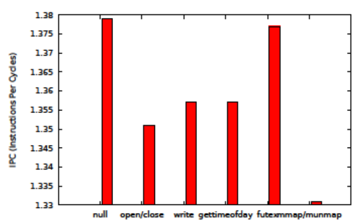


Fig. 2. IPC (Instructions Per Cycles) by six system calls

The Figure 2 shows the effects of six system calls. We did the experiments with the same workloads which dispatch periodically each other system calls. We measured the instructions per cycles of only user-mode, so we predicted that the results should have no difference in all experiments using each other system calls. However, as you can see, the results are significantly different due to the pollutions on the user-mode context by the system calls. Typically the case of *mmap/munmap* decreases the performance about 4%.

3 Related Works

There are many different user-level thread management schemes. In this section, we briefly describe the three schemes which have each different goals.

The scheduler activations [9] scheme is the traditional M:N model thread management scheme which maps some N number of user-level threads onto some M number of kernel-level threads. They achieved the performance and flexibility advantages based on M:N thread model which is implemented using a new kernel interface and user-level thread package, but they may have some stability and security risk due to close communication between user-level thread package and kernel [9].

The Lithe uses also user-level thread model but the goal is to arbitrarily compose parallel libraries without sacrificing performance. They proposed a new unvirtualized thread concept for this goal. The pthread library generally uses virtualized threads which have no steady physical processor and should require a physical processor competitively. On the other hand, the unvirtualized threads have a one-to-one mapping to a physical processor using the processor affinity, and they schedule efficiently user-level threads of each other parallel libraries using physical processing engines on the unvirtualized threads [10].

The FlexSC using exception-less system calls was proposed to reduce the pollution effects on processor structures such as processor caches by traditional synchronous system calls. The exception-less system calls delay the execution of a series of system calls unlike the synchronous system calls, and executes them in a batch on a different core with the one on which the system call was invoked for minimizing the frequency of mode-switch between user and kernel. They evaluated the Apache, MySQL and BIND which are typical server programs, and they improved the performance of them using the FlexSC based on the exception-less system calls [11].

4 MyUT

In this paper, we proposed new user-level thread management scheme for the efficient cache utilization which can replace the pthread library. Our goal is to develop the general-purpose user-level thread management scheme for three workloads. To achieve this goal, we used three schemes: a) no-contention scheduling policy, b) unvirtualized thread and c) exception-less system call. In the following subsections, we explain the architectures and the schemes of MyUT in detail.

4.1 Unvirtualized Thread

The MyUT uses the unvirtualized thread in order to reduce the unnecessary contention about the physical processors. Basically, as you can see the Figure 4, the MyUT creates the one unvirtualized thread for each physical processor that is a kernel-level thread, and the MyUT may or may not set the processor affinity using *pthread_attr_setaffinity_np* for the all unvirtualized threads. The MyUT can reduce the context-switch between the kernel-level threads and the CPU-migration between the physical processors by reducing the number of the kernel-level threads using the unvirtualized thread.

4.2 No-Contention Scheduling Policy

The pthread library on linux uses the kernel-level threads due to the advantages such as stability, security and efficient blocking IO handling. In addition, the light-weight process of the linux kernel helps the pthread library more efficient even though using the kernel-level threads. However, the all threads created by the pthread library should have a contention for the processors under a time-sharing policy. So, the extra overhead due to context-switch is increasingly significant where more threads are run. On the other hand, the MyUT based on the user-level threads uses the no-contention scheduling policy by the user-level scheduler. This scheduling policy is very simple. The all user-level threads using this policy do not be scheduled except when the thread is completed or dispatches system calls. So, this policy doesn't allow all user-level threads to use an infinite loop. The Figure 3 shows briefly the difference of the scheduling policy between the pthread library and MyUT.

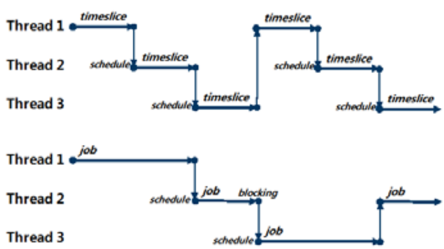


Fig. 3. Scheduling policy of Pthread and MyUT: the upper graph shows the scheduling policy of the pthread, and the bottom graph shows the scheduling policy of the MyUT.

This policy can not be adjusted by the kernel-level scheduler because the kernel-level scheduler can not know the workloads and the characteristics of the each threads. However, the user-level scheduler manages only the owned threads of the specific process, so the user-level scheduler know all of the informations of its threads and can adjust the effective policy to them such as the no-contention scheduling policy.

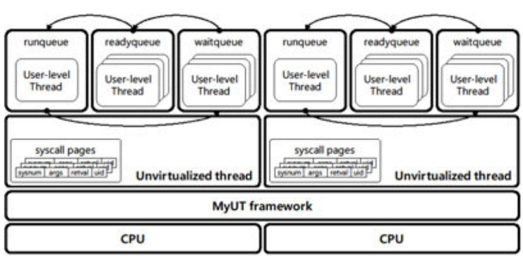


Fig. 4. The architecture of MyUT

The Figure 4 presents how the MyUT based on the no-contention scheduling policy manages user-level threads. The MyUT has one unvirtualized thread per each processor, and the each unvirtualized thread has three queues, **runqueue**, **readyqueue** and **waitqueue**. The **runqueue** manages all of the runnable user-level threads, but it has just one user-level thread running on the unvirtualized thread, because the unvirtualized thread should run the user-level thread until it returns the unvirtualized thread to the other user-level threads by its self. The **readyqueue** manages all of the runnable user-level threads which are waiting for the user-level thread on the **runqueue**. If the user-level thread on the **runqueue** is completed or blocked by system call, the user-level scheduler will choose the first user-level thread on the **readyqueue**. And the **waitqueue** manages the user-level threads which are blocked by system call or locking mechanism. So, if you create a user-level thread, first the user-level thread is enqueued into the **readyqueue**. And sometime after all prior threads completing or blocking, the user-level thread is dequeued from the **readyqueue** and enqueued into the **runqueue**. If the user-level thread dispatches a system call or requests a lock, the user-level thread will be enqueued into the **waitqueue**, and then after completion of the requests, the user-level thread is re-enqueued into the **readyqueue**. These are the normal life cycles of the user-level thread on the MyUT.

4.3 Exception-Less System Call

As we explained the Section 1.2, the synchronous system call is the one of the most important source of the cache pollutions. So, for reducing the overhead of the synchronous system call, the MyUT employs the exception-less system call. The unvirtualized thread on the MyUT has each owned syscall page. The each entries of the syscall page include system call number, arguments, return value and the user-level thread ID. We intercept the system call wrapper functions and replace calling the software interrupt with inserting the corresponding system call informations into the syscall page. And if the unvirtualized thread has no runnable user-level thread, it will dispatch the all pending system calls on the syscall page at a time. This is useful due to reduce the cache pollutions by frequent mode-switch between kernel and user.

However, when the shared caches and branch prediction tables between the user-level threads are very small and the each user-level threads have a lot of private caches, sometimes the cache pollutions by the user-level context-switch are bigger than the cache pollutions by the mode-switch. We verified and analyzed this problem through the experiment in the next section.

And the exception-less system call has the other problem of blocking requests. The Figure 5 shows the scenario of handling multiple system calls on the pthread, FlexSC and MyUT. The pthread uses one kernel-level thread per one user-level thread, so if one system call handler is blocked, the other system call handler can be executed directly on the other kernel-level thread. On the other hand, the FlexSC executes multiple system calls in a batch on the specific syscall thread, so if one system call handler is blocked, the other system call handler should wait until the blocked system call handler is completed. For this reason, the FlexSC creates the several syscall threads per the syscall page, but the fundamental problem still remains. We proposed the **Nested Waiting State (NWS)** scheme to solve this problem. The **NWS** allows the kernel-level thread to handle directly next system call handler after the previous system call blocking. And after the prologue of all system calls is completed, the kernel-level thread will wait the all response of the blocked system calls at once. The performance models of the pthread, FlexSC and MyUT look like follows:

$$T(P)_i = \sum_{k=1}^{i}(P_k + E_k) + W_c + O_c$$

$$T(F)_i = \sum_{k=1}^{i} (P_k + W_k + E_k)$$

$$T(M)_i = \sum_{k=1}^{i} (P_k + E_k) + W_i$$

The $T(P)_i$, $T(F)_i$ and $T(M)_i$ that i means the number of system calls requesting services show respectively the performance models of the pthread, FlexSC and MyUT. The $T(x)_i$ means the total time of handling the multiple system calls. The P_k and E_k mean respectively the time of prologues and epilogues of the system call handler, and the W means the waiting time of the blocked system call, and the O_c means the overhead of additional context-switch due to system call handler blocking. The $T(P)_i$, $T(F)_i$ and $T(M)_i$ shows respectively the performance models of the pthread, FlexSC and MyUT. The $T(P)_i$ which uses one kernel-level thread per one system call and $T(M)_i$ with the **NWS** do not include the summation of waiting times for all blocked system calls due to overlap the waiting periods with the prologue and epilogue periods, moreover the $T(M)_i$ do not include O_c due to handle the multiple system calls on the specific single kernel-level thread.

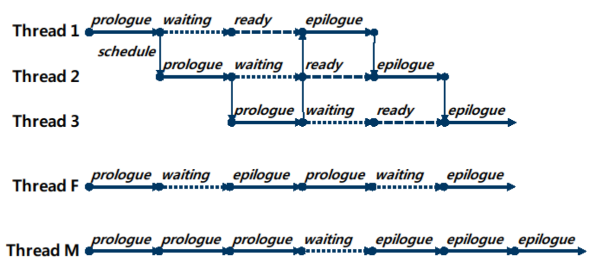


Fig. 5. Multiple system calls handling of Pthread, FlexSC and MyUT: the prologue means the codes which are executed before the system call blocking, and the epilogue means the codes which are executed after the system call waking up

5 Evaluations

Our experimental environment is 2.8 GHz Intel i7 QuadCores. Each processor has 512 KB private L2 cache and 8 MB shared L3 cache with 64 bytes cache line. The operating system is Linux 2.6.33 kernel with Ubuntu 10.04 and the compiler is gcc 4.4.

We first evaluated the number of context-switches and CPU-migrations of the pthread and MyUT. The Figure 6 shows the result of the microbenchmark which performs the prime calculation on the multiple threads. As you can see, the more number of threads run in parallel, the more count of context-switches and CPU-migrations are occurred. The increasing amounts of the pthread are almost twice bigger than the MyUT. Because in the case of the pthread, the number of threads is same with the number of kernel-level threads, but the MyUT has only the four kernel-level threads regardless of the number of threads. So, the MyUT has the smaller count than the context-switches of the pthread, and the count of CPU-migrations do not change almost whether the number of threads is larger or smaller.

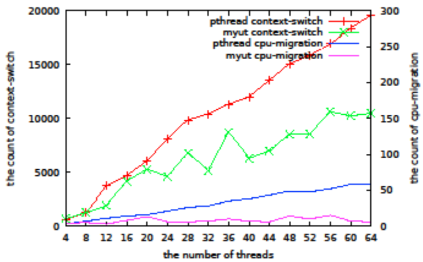


Fig. 6. Context-switch and CPU-migration of Pthread and MyUT

The next experiment includes three our microbenchmarks, **prime**, the prime calculation for CPU-intensive workload, **mem**, the memory access for memory-intensive workload, and **sysc**, the system call dispatch for IO-intensive workload on the pthread and MyUT. The Table 1 shows the count of context-switches, CPU-migration, cache-misses and branch-misses of each benchmarks. We observed the cache-misses and branch-misses improvements of up to 19% in the cases of **prime** and **mem** which consume a lot of caches and branches prediction table, but the **sysc** which dispatches only **write** system call repetitively is improved up to 9%. And the MyUT with work-stealing is better performance than the other cases because the unvirtualized thread on the underloaded processor steals the jobs of the unvirtualized thread on the overloaded processor using the work-stealing scheme. On the other hand, in the case of the MyUT with processor-affinity, the unvirtualized thread on the overloaded processor can not migrate to the other processors, so the total performance shows worse than the others.

Table 1. The cost of Prime Calculation and Memory access (w: with work-stealing, p: with processor-affinity): the numbers in the brackets denote the proportion of the proposed schemes to the pthread for each cases

workload	*type*	*context-switch*	*CPU-migration*	*caches-misses*	*branch-misses*
	pthread	18642	60	84217	6044611
	myut	10121	31	67841	5777262
prime	myut-w	10110	8	68660	5770601
	myut-p	10332	10	72200	5794377
	myut-wp	10163	8	71964	5794377
	pthread	39119	211	149044	3601229
	myut	20148	13	126126	3122027
mem	myut-w	20396	13	127356	3140520
	myut-p	20323	11	128809	3154012
	myut-wp	20258	15	129318	3127874
	pthread	35648	198	187092	13492779
	myut	18841	23	175890	12392326
sysc	myut-w	18921	11	176371	12545469
	myut-p	18688	26	181823	12485606
	myut-wp	18951	10	177363	12344309

The last experiment shows the benchmark about the user-level threads which have owned private buffer on the pthread and MyUT and access once all private buffer between each system calls. We thought if the user-level threads have the amount of private buffer, the overhead of the user-level scheduling by the exception-less system call is bigger than the overhead of the synchronous system call. Because the synchronous system call causes the cache pollutions by the system call handler only, but the exception-less system call causes the cache pollutions by the other user-level threads, so if the private buffers of the user-level threads are big, the cache pollutions by the other user-level threads can be more than that by the system call handler. The Figure 7 shows the results about that. Our experimental quad-core processor has 512 KB private L2 cache and 8 MB shared L3 cache, and each cores can use almost 2 MB processor cache. So, when the size of private buffer is enough small, the MyUT with the exception-less system call shows better performance than the pthread with the synchronous system call, because the cache conflicts between the private buffer of the user-level threads are enough small. However, when the size of private buffer is bigger than 1024 KB, the cache conflicts of the MyUT increase nonlinearly due to the amount of the user-level scheduling which are occurred by every system calls. We verified that the exception-less system call shows worse performance than the synchronous system call in the case of the user-level threads which have the amount of private buffer through this experiment.

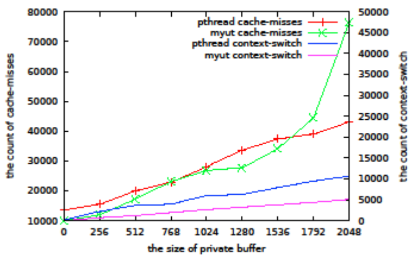


Fig. 7. The pollutions on private buffer by system calls of Pthread and MyUT

6 Conclusion

In this paper, we proposed the general purpose user-level thread management scheme, and evaluated the three workloads using the proposed scheme. Through these experiments, we analyzed and found how the three policies and techniques affect the cache utilization for the workloads which are divided into the user-level threads. As the future works, we will apply the proposed scheme to practical and composite workloads, and through the results we will optimize and improve our user-level thread management scheme.

Acknowledgments. This research was supported by Future-based Technology Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (2010-0020730).

References

1. Fluet, M., Rainey, M., Reppy, J.: A Scheduling Framework for General-Purpose Parallel Languages. In: International Conference on Functional Programming, Victoria, British Columbia, Canada (September 2008)

2. Liu, R., et al.: Tessellation: Space-time Partitioning in a ManyCore Client OS. In: HotPar, Berkeley, CA (2009)
3. Fromm, R., Treuhaft, N.: Revisiting the Cache Interference Costs of Context Switching, http://citeseer.ist.psu.edu/252861.html
4. Zhang, X., Dwarkadas, S., Shen, K.: Towards Practical Page Coloring-based Multicore Cache Management. In: EuroSys, Nuremberg, Germany (2009)
5. Li, C., Ding, C., Shen, K.: Quantifying the Cost of Context Switch. In: ExpCS, San Diego, CA (2007)
6. Mogul, J.C., Borg, A.: The Effect of Context Switches on Cache Performance. In: The Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, Santa Clara, CA (1991)
7. Agarwal, A., Hennessy, J.L., Horowitz, M.: Cache Performance of Operating System and Multiprogramming Workloads. ACM Trans. on Computer Systems 6(4), 393–431 (1988)
8. Tullsen, D., et al.: Exploiting Choice: Instruction Fetch and Issue on an Implementable Simultaneous Multithreading Processor. In: ISCA, Philadelphia, PA (1996)
9. Anderson, T.E., et al.: Scheduler Activations: Effective Kernel Support for the User-level Management for Parallelism. In: SOSP, Pacific Grove, CA (1991)
10. Pan, H., Hindman, B., Asanovi, K.: Composing Parallel Software Efficiently with Lithe. In: PLDI, Toronto, Canada (2010)
11. Soares, L., Stumm, M.: FlexSC: Flexible System Call Scheduling with Exceptionless System Calls. In: OSDI, Vancouver, Canada (2009)

Relay Selection with Limited Feedback for Multiple UE Relays

Kyungrok Oh and Dong In Kim*

School of Information and Communication Engineering, Sungkyunkwan University
Suwon 440-746, Korea
{yob00,dongin}@skku.edu

Abstract. When a source has insufficient information about relay-to-destination (R-D) link, like type-2 user equipment (UE) relaying, using average channel gain severely degrades the end-to-end error performance. In this paper, we propose a threshold based relay selection with limited feedback, which has more specific information about R-D link. For a simple case, we consider two adaptive thresholds, where the R-D channel gain is quantized to three levels. Candidate relays whose signal strength exceeds a threshold are divided into two groups according to the threshold, and a source selects the best relay considering the modified harmonic mean of channel gains. We suggest the method of finding optimal thresholds and present the symbol-error rate (SER) performance of M-ary PSK signaling. Simulation results show that relay selection with limited feedback can improve the SER performance, especially in asymmetric link conditions.

Keywords: Cooperative relaying, relay selection, limited feedback, adaptive threshold, symbol-error rate (SER).

1 Introduction

Wireless communications enable nodes to transmit information to others by the nature of broadcasting. However, when nodes are distant from the source node, low Signal-to-Noise Ratio (SNR) causes a link failure. Such problem can be reduced by using multiple antennas that enhance the communication data rate [1]. Another solution is using cooperative communications to achieve the spatial diversity since making a device with multiple antennas is difficult. In cooperative communications, relay nodes help transmit information from the source to destination as a virtual antenna array.

Different cooperative communication schemes were suggested and studied as shown in [2], [3], and [4]. Amplify-and-forward (AF) relaying and decode-and-forward (DF) relaying are popular cooperation protocols to achieve the spatial

* This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency" (NIPA-2011-(C1090-1111-0005)).

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 157–166, 2011.

diversity. In [5], the authors derived a symbol error rate (SER) for the multi-relay DF protocol. In [6], they proposed the relay selection protocols using a harmonic mean of multi-relay communication, based on the SER performance. In [7], the authors suggested the 1-bit feedback relay selection scheme using an adaptive threshold, which was calculated by the SER.

In an open loop state, when a source has no information of relay to the destination link, feedback information can notify the channel states to the source. In [7], the best relay is selected by referring to both the S-R (source to relay) link and R-D (relay to destination) link conditions. The authors proposed the best relay selection method for choosing the best S-R link when its R-D channel gain exceeds a pre-determined threshold. However, using this 1-bit feedback information, we might have the possibility of losing the better R-D channel links since we only have the information of availability of the R-D links. For this reason, when we use more specific feedback information, like 2-bit feedback, the performance can be improved. Increasing the feedback information can raise the complexity of system, and hence we consider the 2-bit feedback relaying scheme with a slight increase in system complexity.

In this paper, we suggest threshold based limited feedback relay selection for DF protocol in wireless networks. We quantize the R-D channel gain with three levels using the SER formulation of DF protocols. We optimize the adaptive threshold to minimize the SER, and the threshold varies with the system overall SNR. In case of the limited feedback, when the number of cooperating relay nodes is increased, the effect of specific feedback information is decreased and quite similar to 1-bit scheme. However, in a practical case (asymmetric link conditions and cooperating relays N=2), limited 2-bit feedback scheme provides better performance than 1-bit scheme. It is found that more feedback information assures better R-D channel condition for the best relay, resulting in an improved SER performance. We derive the SER performance under both the symmetric and asymmetric channel conditions. We expect that this limited feedback scheme will yield a good performance, especially for the asymmetric type-2 UE relay communications.

The rest of this paper is organized as follows. In Section 2, we present the system model and assumptions for cooperative multiple relays. Section 3 describes the proposed DF relaying strategies with 2-bit feedback information. We present the simulation results to verify the advantages of using limited feedback in Section 4, and finally conclusion is given in Section 5.

2 System Model

Throughout this paper, we consider cooperative wireless network with a source, a destination and N other relay nodes, as shown in Fig. 1. In this figure, S represents a source, R is the relay nodes, and D is a destination. We adopt decode-and-forward (DF) protocol and consider a cooperation strategy with two phases (half-duplex mode). In the first phase, a source broadcasts its signal to

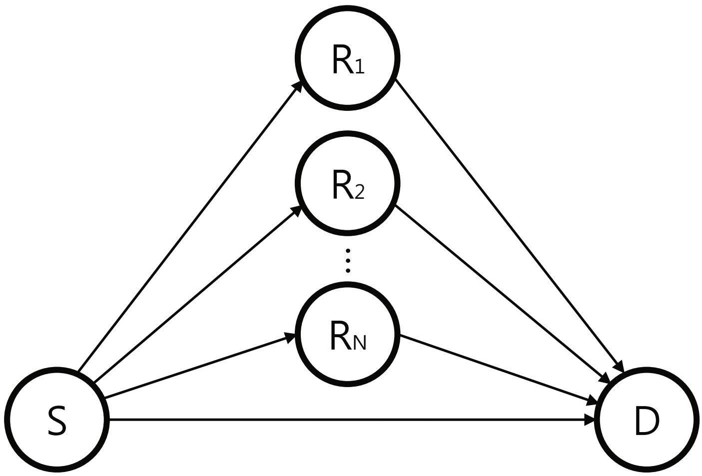


Fig. 1. The wireless network model with N relays

the relays and a destination. The received symbol at the destination and the relay nodes can be expressed as

$$y_{s,d} = \sqrt{P_1}h_{s,d}x + n_{s,d} \tag{1}$$

$$y_{s,r_i} = \sqrt{P_1}h_{s,r_i}x + n_{s,r_i} \tag{2}$$

where P_1 is the source transmit power, x is the transmitted signal, $h_{s,d}$ and $h_{s,r}$ are the channel gains of S-D and S-R_i links, respectively, and $n_{s,d}$ and n_{s,r_i} are additive white Gaussian noises (AWGNs).

The selected relay forwards the received data to the destination in the second phase only when the relay decodes the data correctly. If the relay fails to decode the received data, it remains idle. The received signal at the destination can be expressed by

$$y_{r_i,d} = \sqrt{\tilde{P}_2}h_{r_i,d}x + n_{r_i,d} \tag{3}$$

where x is the transmitted signal from the selected relay, $h_{r_i,d}$ is the channel gain of R_i-D link, $n_{r_i,d}$ is AWGN. If the selected relay decodes correctly, then $\tilde{P}_2 = P_2$ and otherwise $\tilde{P}_2 = 0$. The total power of the system is fixed to $P = P_1 + P_2$, where P is the available maximum transmit power. We assume that the destination employs the maximal-ratio combining (MRC) for the received signals from both the source and the relay. The output of the MRC is given by

$$y = \frac{\sqrt{P_1}h_{s,d}}{\sigma_n^2}y_{s,d} + \frac{\sqrt{\tilde{P}_2}h_{r_i,d}}{\sigma_n^2}y_{r_i,d}. \tag{4}$$

We assume flat quasi-static channel state so that the channel gains remain constant over the time slot and are independent from slot to slot. We also assume

Time Division Duplex (TDD) mode in the relay selection where $h_{s,d}$ and h_{s,r_i} are known at the source and $h_{r_i,d}$ is known at the relay. Hence, the source has no information about $h_{r_i,d}$, it has to achieve the status of $h_{r_i,d}$ through the feedback information. For instance, 2-bit feedback information can be used to notify the source of the channel gains with 3 quantized levels.

The coefficients of the channel gains $h_{s,d}$, h_{s,r_i}, and $h_{r_i,d}$ are modeled as zero-mean complex Gaussian random variables with variances $\delta^2_{s,d}$, δ^2_{s,r_i}, and $\delta^2_{r_i,d}$, respectively. The absolute value of the channel coefficient $|h_{x,y}|$ is Rayleigh distributed, and $|h_{x,y}|^2$ is exponentially distributed with mean $\delta^2_{x,y}$. We model the noise variables $n_{s,d}$, n_{s,r_i}, and $n_{r_i,d}$ as zero-mean complex Gaussian random variables with variance σ^2_n.

3 Selection Strategies

In [7], authors proposed the adaptive threshold minimum feedback and they showed that only 1-bit information can achieve a quite good performance. The mechanism of proposed 1-bit feedback is 1) find whether the R-D link is available or not by comparing with the calculated threshold and then, 2) select the best S-R link. To extend the proposed scheme, we consider the limited feedback which has more specific information about the R-D link. Since 1-bit feedback can only find the available link conditions with minimum threshold channel gain, it is likely to miss some links with better channel conditions. Therefore, if we have more precise information about the channel conditions, we can further improve the SER performance.

If we can use the instantaneous channel gains of S-R and R-D links for relay selection, the harmonic mean function of these channel gains guarantees the better SER performance. In the limited feedback system, we may also use the harmonic mean function for the S-R links when their R-D channel gains exceed given thresholds. The harmonic mean function of S-R and R-D channel gains [6] is defined as

$$\beta_i = \frac{2q_1q_2\beta_{r_i,d}\beta_{s,r_i}}{q_1\beta_{r_i,d} + q_2\beta_{s,r_i}} \quad (i = 1, 2, \cdots, N), \tag{5}$$

where $q_1 = \frac{A^2}{r^2}$, $q_2 = \frac{B}{r(1-r)}$, and $\beta_{x,y} = |h^2_{x,y}|$. In addition, A, B, and r, the calculated factors of SER for M-PSK signaling can be defined as

$$A = \frac{M-1}{2M} + \frac{\sin\left(\frac{2\pi}{M}\right)}{4\pi} \tag{6}$$

$$B = \frac{3(M-1)}{8M} + \frac{\sin\left(\frac{2\pi}{M}\right)}{4\pi} - \frac{\sin\left(\frac{4\pi}{M}\right)}{32\pi} \tag{7}$$

$$r = \frac{P_1}{P}. \tag{8}$$

The basic idea of the proposed limited feedback scheme is to quantize the R-D link with three levels, and then select the relay which maximizes the harmonic mean in (5). The quantized R-D channel gains value can be written as

$$\beta_{r_i,d} \in \{0, \beta_{th1}, \beta_{th2}\}. \tag{9}$$

For example, if we consider the limited feedback like 2-bit, there can be three thresholds except zero. In this paper, we adopt only two thresholds for the simplicity. The source can define the "good" and "better" status of the R-D channel gains by feedback information. Since the two thresholds are some fixed values for each SNR, choosing the maximum S-R link from the two candidates will select the best relay. To obtain the optimal values, we define one candidate whose R-D channel gain exceeds the threshold β_{th1} as

$$\beta_a = \max\{\beta_{s,r_1}, \beta_{s,r_2}, \cdots, \beta_{s,r_{k_1}}\} \tag{10}$$

and the other candidate whose R-D channel gain exceeds the threshold β_{th2} is

$$\beta_b = \max\{\beta_{s,\tilde{r}_1}, \beta_{s,\tilde{r}_2}, \cdots, \beta_{s,\tilde{r}_{k_2}}\} \tag{11}$$

where $k_1 + k_2 = N$. Consequently we only have to choose the best S-R link from the two candidates to maximize (5). We can use both β_{th1} and β_a for the first candidate. Also, β_{th2} and β_b can be adjusted for the second candidate.

$$\beta_{k_1^*} = \frac{2q_1q_2\beta_{th1}\beta_a}{q_1\beta_{th1} + q_2\beta_a} \tag{12}$$

$$\beta_{k_2^*} = \frac{2q_1q_2\beta_{th2}\beta_b}{q_1\beta_{th2} + q_2\beta_b}. \tag{13}$$

The maximized channel gain can be written as

$$\beta = \max\{\beta_{k_1^*}, \beta_{k_2^*}\}. \tag{14}$$

To find the optimal threshold β_{th1} and β_{th2}, we can use the SER formulation in [6] and the probability density function (pdf) of each link channel gain. The continuous pdf of the R-D link gain is converted to the discrete pdf because of the quantized thresholds. For the S-D link, the pdf of its channel gain is exponential and can be expressed as

$$p_{s,d}(\beta) = \frac{1}{\delta_{s,d}^2} e^{-\beta/\delta_{s,d}^2}. \tag{15}$$

For the S-R link, the cumulative distribution function (CDF) of β_a and β_b which have K possible relays can be written as

$$\begin{aligned} P_{\beta_a}(\beta|K = k_1) &= Pr(\beta_{s,r_1} \le \beta, \beta_{s,r_2} \le \beta, \cdots, \beta_{s,r_{k_1}} \le \beta) \\ &= (1 - e^{-\beta/\delta_{s,r}^2})^{k_1}, \end{aligned} \tag{16}$$

$$P_{\beta_b}(\beta|K=k_2) = Pr(\beta_{s,\tilde{r}_1} \leq \beta, \beta_{s,\tilde{r}_2} \leq \beta, \cdots, \beta_{s,\tilde{r}_{k_2}} \leq \beta) \\ = (1 - e^{-\beta/\delta^2_{s,r}})^{k_2}. \quad (17)$$

By differentiating the CDF, we can obtain the pdf of β_a and β_b as

$$p_{\beta_a}(\beta) = \frac{dP_{\beta_a}(\beta)}{d\beta}, \quad (18)$$

$$p_{\beta_b}(\beta) = \frac{dP_{\beta_b}(\beta)}{d\beta}. \quad (19)$$

For the R-D link, the discrete pdfs of β_{th1} and β_{th2} can be calculated as constant P_a and P_b, respectively. Then, the SER can be formulated as

$$Pr(e) = Pr(e/\psi)Pr(\psi) + Pr(e/\psi^c)Pr(\psi^c) \quad (20)$$

where $Pr(e/\psi)Pr(\psi)$ denotes the SER of the direct transmission and $Pr(e/\psi^c)$ $Pr(\psi^c)$ represents the SER of the relay cooperation. We can calculate the closed-form β_a and β_b using the given pdf and SER. However, due to the limited space we only explain the main concepts and omit the detailed procedure of deriving an exact form.

4 Simulation Results

We consider the modulation order $M = 4$ in PSK signaling. Also, we assume the simulation models as described in Section 2. Using this simulation, we choose the two thresholds by obtaining the quasi-optimal thresholds, which minimize the SER. For fair comparison, the SER performance is plotted. An optimum power allocation is not addressed because we perform the simulation assuming the equal power of $P_1 = P_2$. We assume the unity channel variance, where $\delta^2_{s,d}$ $= \delta^2_{s,r} = \delta^2_{r,d} = 1$ in symmetric cases and the noise variance is set to $\sigma^2_n = 1$.

In Fig. 2, we evaluate the effect of adaptive thresholds and feedback bits on the SER performance. For that, we adopt the fixed thresholds of β_{th1} and β_{th2} when $P/\sigma^2_n = 15$dB, and the SER performances with adaptive thresholds and fixed thresholds are compared. We first show the results under the symmetric case. It is shown that using the fixed thresholds yields poor performance, so the pre-defined adaptive thresholds are preferred. Moreover, for the 1-bit feedback case with fixed threshold β_{th1} at the same SNR, it also yields poor performance in the high SNR region.

Fig. 3 shows the SER performances of 2-bit limited feedback, 1-bit feedback, instantaneous harmonic mean and average channel gain under the symmetric channel conditions. As expected, the instantaneous harmonic mean case has the best SER performance. However, this scheme can be used in a closed-loop system where the source knows about all instantaneous channel gains. We use (5) to obtain the results for the instantaneous harmonic mean and limited feedback

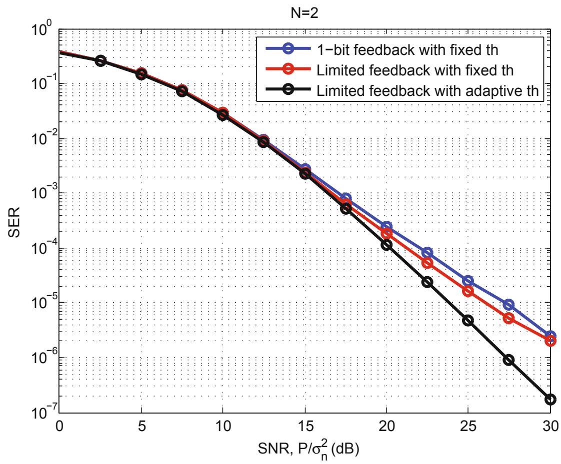


Fig. 2. SER comparison with adaptive threshold and fixed threshold at $P/\sigma_n^2 = 15$dB when N=2

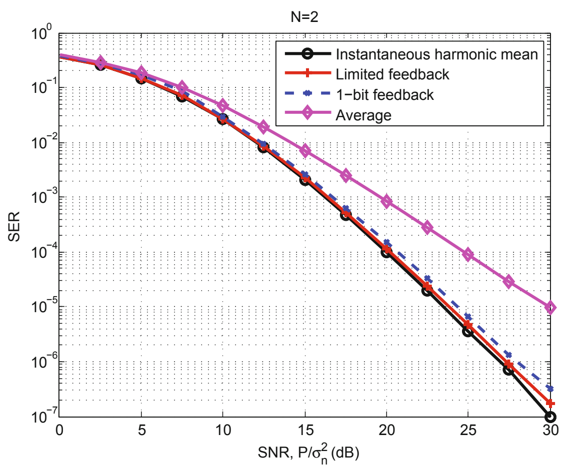


Fig. 3. SER comparison with other schemes under symmetric link conditions

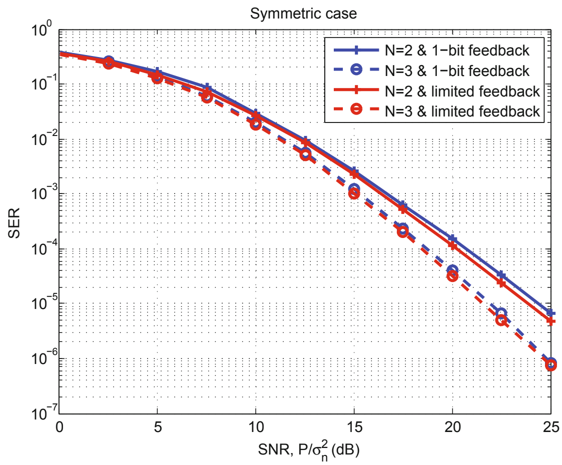


Fig. 4. SER comparison with limited feedback and 1-bit feedback under symmetric link conditions

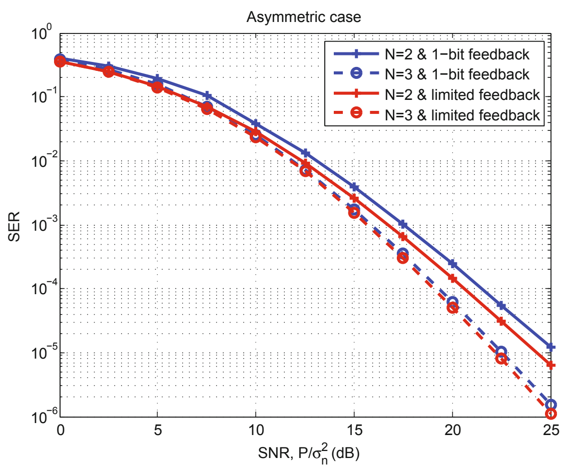


Fig. 5. SER comparison with limited feedback and 1-bit feedback under asymmetric link conditions

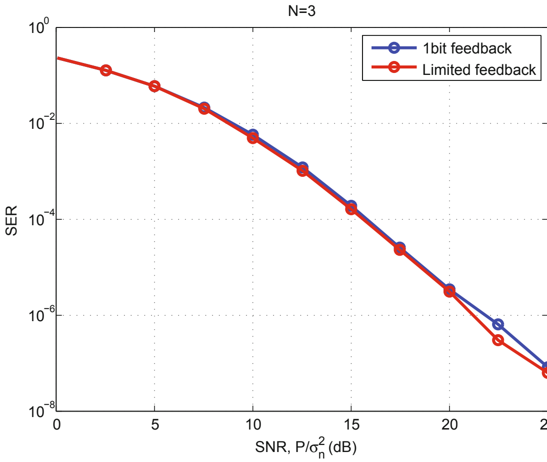


Fig. 6. The SER simulated with fixed thresholds, when $R_i - D$ average channels are stronger than $S - R_i$

cases as a selection metric. The worst case of using the average channel gain is also plotted. In this case, the source only has the information of $R_i - D$ average channel gain $\delta^2_{r_i,d}$ instead of the instantaneous channel gains $\beta_{r_i,d}$. Meanwhile, we see that the proposed limited 2-bit feedback scheme achieves better performance than the 1-bit feedback case.

Fig. 4 depicts the SER performance under the symmetric link conditions. As the number of relays increases, the SER performance is getting improved. The performance is quite acceptable in most of the interesting SNR range. In the symmetric cases, the performance of the limited 2-bit feedback is better than the 1-bit feedback case. As the number of relay nodes increases, the performance gap between them is decreased. However, the performance gap can be increased if we optimize the power allocation between the source and relays with total power constraint and also the optimal thresholds more accurately.

In Fig. 5, we plot the SER of the asymmetric case for two and three relays. For $N = 2$ relays, we set the average channel gain as $\delta^2_{s,r_1} = \delta^2_{r_2,d} = 0.5$ and $\delta^2_{s,r_2} = \delta^2_{r_1,d} = 1.5$. For $N = 3$ relays, the first two relays have the same average channel gains as in $N = 2$, and the other is set to $\delta^2_{s,r_3} = \delta^2_{r_3,d} = 1$. For comparison with the symmetric cases, we assume the equal mean values of S-R and R-D average channel gains for both symmetric and asymmetric cases. We can see that the asymmetric case with $N = 2$ provides some gain over the symmetric case with $N = 2$. For $N = 3$, the two cases perform similarly in the SER performance since the effective diversity gain dominates the feedback information gain.

Fig. 6 shows the SER performance when $R_i - D$ average channel gains are relatively stronger, namely $\delta^2_{s,r_1} = \delta^2_{s,r_2} = \delta^2_{s,r_3} = 10$ and $\delta^2_{s,r_1} = \delta^2_{s,r_2} = \delta^2_{s,r_3} = 1$. The SER curve reveals that using limited 2-bit feedback yields better performance than the 1-bit case for fixed thresholds. This is because if we have no information about the optimal thresholds, using more feedback information improves the SER performance, especially when the $R_i - D$ links are in better conditions. Therefore, the proposed scheme will be suitable for the multiple relays network where the relatively better R-D channel gains are guaranteed.

5 Conclusion

We have proposed the relay selection scheme with limited feedback information on R-D link quality in the DF multiple relay networks. We considered wireless cooperative network consisting of one source, N possible relay nodes, and one destination, where the power of S-R and R-D links was kept the same and QPSK signaling was employed for simplicity. We have tested the feasibility of limited feedback protocol through the simulation analysis. The one-bit limited feedback protocol was shown in [7] to track the performance of instantaneous channel gain based protocol well, and in this work we showed that having 2-bit information on the R-D channel gain provides the SER performance very close to that of the instantaneous case with a slight increase in system complexity. We also compared the symmetric and asymmetric cases with varying channel gain variances, and in all cases, the 2-bit feedback scheme improved the SER performance when compared to the 1-bit case. Especially, the efficiency of having more feedback information was clearly observed under the asymmetric channel conditions. In our future work, we will derive a closed-form SER performance for the limited feedback scheme while optimizing the optimal thresholds analytically.

References

1. Foschini Jr., G.J., Gans, M.J.: On limits of wireless communication in a fading environment when using multiple antennas. In: Wireless Personal Commun., pp. 36–54 (1998)
2. Laneman, J.N., Tse, D.N.C.: Cooperative diversity in wireless networks: efficient protocols and outage behavior. IEEE Trans. Inform. Theory. 50, 3062–3080 (2004)
3. Su, W., Sadek, A.K., Liu, K.J.R.: SER performance analysis and optimum power allocation for decode-and-forward cooperation protocol in wireless networks. In: Proc. IEEE Wireless Communications and Networking Conference, vol. 2, pp. 984–989 (2005)
4. Kramer, G., Gastpar, M., Gupta, P.: Cooperative strategies and capacity theorems for relay networks. IEEE Trans. Inf. Theory 51(9) (2005)
5. Fareed, M.M., Uysal, M.: On relay selection for decode-and-forward relaying. IEEE Trans. Wireless Commun. 8(7) (2009)
6. Ibrahim, A.S., Sadek, A.K., Su, W.: Cooperative communications with relay-selection: when to cooperate and whom to cooperate with? IEEE Trans. Wireless Commun. 7(7) (2008)
7. Park, S.C., Kim, D.I.: Adaptive threshold based relay selection for minimum feedback. In: Proc. CHINACOM 2010, Beijing, China, pp. 1–5 (2010)

Enhanced Multi-homing Support for Proxy Mobile IPv6 in Heterogeneous Networks

Dae Sun Kim, Yoshinori Kitatsuji, and Hidetoshi Yokota

KDDI R&D Laboratories, Inc. Fujimino, Saitama, Japan,
{da-kim,kitaji,yokota}@kddilabs.jp

Abstract. The dynamic nature of future network environments, where a huge number of nodes of diverse types and equipped with a variety of access technology interfaces will be dynamically moved across heterogeneous networks, calls for an enhanced mobility function that facilitates multi-homing support. Therefore, in this paper, we analyze the lack of multi-homing support in the Proxy Mobile IPv6 specification that has become the standardized mobility management protocol in 3GPP (SAE/LTE) and WiMAX Forum (NWG), then propose an enhanced multi-homing support scheme based on multiple IPv6 address allocation to maintain application session continuity between two interfaces of a mobile node during an inter-technology handover. The proposed scheme is validated using QualNet simulator.

Keywords: Proxy Mobile IPv6, Multi-Homing, Session Migration, Heterogeneous Networks, Randomized Interface Identifier.

1 Introduction

Mobile networks are continuously evolving by incorporating new technologies. Future mobile networks are therefore expected to include various heterogeneous networks, ranging from macro to micro, pico and even femtocells, with diverse types of mobile nodes equipped with multiple interfaces, because a single-access network cannot provide ubiquitous coverage and a continuously high quality of service-level communications. A mobile node (MN) will have multiple interfaces (a multi-homed MN) with different access technologies to access a variety of networks, thus multi-homing[1][2] which is the technology where a single mobile node has multiple connections to the Internet to increase reliability of network applications, has become essential technology for future networks. Currently, a new network-based mobility-management protocol called Proxy Mobile IPv6(PMIPv6)[3] is being standardized by the IETF and has become the de-facto standard in 3GPP[4] and the WiMAX Forum[5]. However, PMIPv6 has only limited support for multi-homing technology because according to the PMIPv6 specification, each mobility session should be managed under a separate Binding Cache Entry (BCE)[3] at a Local Mobility Anchor (LMA)[3], and thus PMIPv6 simply allows simultaneous access for a multi-homed MN[6]. Therefore, when one interface of a multi-homed MN is detached from its access network, the LMA removes its mobility session from the BCE and the existing

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 167–180, 2011.

flows that are associated with the detached interface are not delivered to the multi-homed MN, although another interface of the multi-homed MN is still connected to another access network. To address this problem, we propose an enhanced multi-homing support scheme to provide seamless mobility between interfaces for a multi-homed MN. The proposed scheme is able to move an application session from a detached interface of a multi-homed MN to an attached interface by using multiple IPv6 address allocation.

The remainder of the paper is organized as follows. Section 2 analyzes the limitation of multi-homing support in PMIPv6. Section 3 describes the additional considerations to support multi-homing in PMIPv6. Section 4 explains the proposed multi-homing support scheme in detail. Simulation results are presented in Section 5. Finally, Section 6 concludes the paper.

2 Analysis of Multi-homing Support in Proxy Mobile IPv6

This section analyzes multi-homing support in PMIPv6. In particular, we highlight the problems, which are the limitation of multi-homing support, the difficulty of implementing the handoff indicator option, and the multiple IPv6 address allocation in PMIPv6.

2.1 Limitation of Multi-homing Support with PMIPv6

Figure 1 shows the simultaneous access when a multi-homed MN enters a PMIPv6 domain.

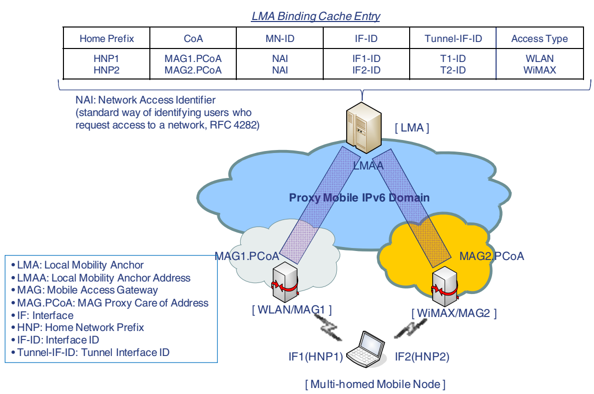


Fig. 1. Simultaneous access through multiple interfaces

The multi-homed MN can access both WLAN and WiMAX simultaneously, as shown in Fig 1. However, when the WLAN interface IF1 of the multi-homed MN is detached from WLAN as shown in Fig 2, the LMA of the Proxy Mobile IPv6 domain removes the mobility session for WLAN interface IF1 from the BCE. Therefore, the existing flows that are associated with HNP1 are no longer delivered to the multi-homed MN under current PMIPv6, although those flows can be delivered to the

WiMAX interface IF2. Therefore, to support enhanced multi-homing in PMIPv6, the LMA needs to group mobility bindings referring to the identical multi-homed MN. In other words, each mobility session for the multi-homed MN should be managed under the same BCE by the LMA.

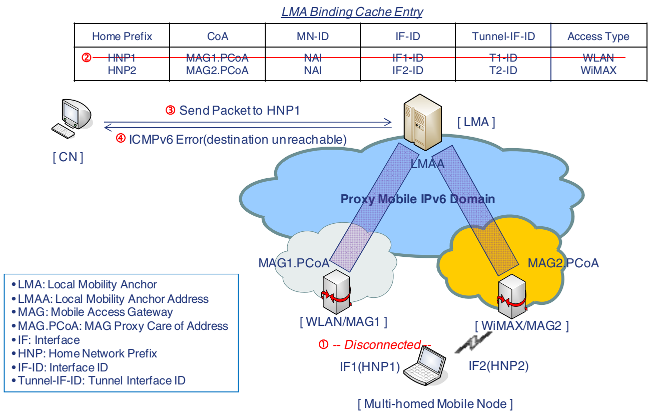


Fig. 2. Limitation of multi-homing support in PMIPv6

2.2 Difficulty of Implementing Handoff Indicator Option

According to the PMIPv6 specification, a new handoff indicator option[3] must be presented in the Proxy Binding Update message[3] and the MN's handoff-related hints between an LMA and a MAG can be exchanged by using Proxy Binding Acknowledgement messages[3]. The significance of the six handoff indicator values is currently defined as shown in Table 1.

Table 1. Six handoff indicators

Value	Description
0	Reserved
1	Attachment over a new interface
2	Handoff between two different interfaces of the mobile node
3	Handoff between mobile access gateways for the same interface
4	Handoff state unknown
5	Handoff state not changed (Re-registration)

The problem is that the value 2 is an unrealistic handoff indicator as of now. There is no way for the MAG to recognize whether the handoff is between two different interfaces of the MN or not since the MN does not participate in the mobility procedure.

2.3 Multiple IPv6 Address Allocation

According to the IPv6 addressing architecture[7], a single interface can be assigned multiple IPv6 addresses. Thus it is possible for an LMA to allocate multiple home network prefixes to each interface of a multi-homed MN and then for the multi-homed MN to generate an IPv6 address by combining the network prefix and the EUI-64 format interface ID[8] that is derived from the MAC address[8]. However, since each interface has a unique MAC address, it is impossible for the multi-homed MN to assign an identical IPv6 address from one interface to another during a hand-over between two interfaces. As a result, the end-to-end connection will be disconnected between the multi-homed MN and the Correspondent Nodes (CNs).

3 Proxy Mobile IPv6 Multi-homing Support Considerations

As we explained in the previous section, there are problems in supporting multi-homing technology with PMIPv6. To address those problems, the core functional entities such as the LMA, MAG and multi-homed MN in a Proxy Mobile IPv6 domain will be described with additional consideration of how to support multi-homing technology.

3.1 Multi-homed Mobile Node Consideration

To solve the address allocation problem that was explained in Section 2.3, a randomized interface identifier[9] can be considered as a way to generate an IPv6 address rather than the EUI-64 format interface ID. If a multi-homed MN uses an identical randomized interface identifier for all of its interfaces, it is possible to assign the IPv6 address that was assigned to one interface to another interface. In this way, the LMA would be able to allocate the identical home network prefix to another interface of the multi-homed MN during the handover period. In this paper, a randomized interface identifier is also used as an interface identifier[3] in a BCE at an LMA and it is conveyed to a MAG by a Router Solicitation message.

3.2 Local Mobility Anchor Consideration

Each mobility session for the multi-homed MN should be managed under the same BCE to support mobility session migration during handover periods. When one interface of the multi-homed MN is detached from its access network, the LMA should be able to forward all flows that are associated with the detached interface to another interface of the multi-homed MN. Thus, it is required that the LMA allocates the home network prefix that was used for the detached interface to the attached interface of the multi-homed MN. To achieve this, we define two new message types, which we call the Mobility Session Migration message and the Mobility Session Migration Acknowledgement message.

3.2.1 Mobility Session Migration Message

This paper defines the Mobility Session Migration (MSM) message that is sent by a Local Mobility Anchor to a Mobile Access Gateway. A new flag (M) is included in the MSM message. The MSM is designated as MH type 10 in the Mobility Header as

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
                                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |          Sequence #           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|M|D| Reserved  |                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                                                               .
.                       Mobility options                        .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fig. 3. Mobility Session Migration message format

defined in RFC 3775[10]. When this value is indicated in the MH Type field, the format of the message data field in the Mobility Header is as follows:

- **Sequence#:** This is a 16-bit unsigned integer used by the node receiving *Mobility Session Migration* and by the sending node to match a returned *Mobility Session Migration Acknowledgement* with this *Mobility Session Migration*.
- **'M' flag:** When this field is set to 1, the MAG must send a *Router Advertisement message* with multiple prefix information to the multi-homed mobile node.
- **'D' flag:** When this field is set to 1, the MAG must send a *Router Advertisement* message with a lifetime value of zero. 'D' flag indicates that this message is a *De-Registration Mobility Session Migration* message.
- **MN-ID option:** This is the Mobile Node Identifier option, which is standardized in RFC 4283[11].
- **Home Network Prefix option:** This option is used for exchanging the multi-homed MN's home network prefix information.

3.2.2 Mobility Session Migration Acknowledgment Message

This paper defines the Mobility Session Migration Acknowledgement (MSMA) message that is sent by a Mobile Access Gateway to a Local Mobility Anchor in response to an MSM message. A new flag (A) is included in the MSMA message. The MSMA is designated MH type 11 in the Mobility Header as defined in RFC 3775. When this value is indicated in the MH Type field, the format of the message data field in the Mobility Header is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
                                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |          Sequence #           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|A|    Reserved |                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                                                               .
.                       Mobility options                        .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Fig. 4. Mobility Session Migration Acknowledgement message format

- **Sequence#:** This is a 16-bit unsigned integer used by the node receiving the *Mobility Session Migration* and by the sending node to match the returned *Mobility Session Migration Acknowledgement* for this *Mobility Session Migration*.
- **'A' flag:** When this field is set to 1, it indicates that the *MSM* was accepted by the MAG.

3.2.3 Extensions to Binding Cache Entry Data Structure

To support multi-homing and inter-technology handover, a Binding Cache Entry Data structure needs to be extended with the following additional fields.

- **Multi-homed Mode field:** This field is a 1-bit field to distinguish whether a mobile node is a multi-homing mode or not. The basic rule is that this field is set to 1 when the LMA receives *Proxy Binding Update* messages with identical NAIs[12] from different MAGs. Otherwise, this field is set to zero.

- **State field:** This is a 1-bit field used by an LMA to signify whether each entry is in the active state or the idle state. If the entry is in the active state, this field set to 1; otherwise, it is set to zero.

- **Active state:** There is incoming/outgoing traffic.
- **Idle state:** There is no incoming/outgoing traffic.
- **Timeout:** After a time limit, an entry indicating the idle state will be removed in the Binding Cache Entry. (Then the LMA sends a Proxy Binding Acknowledgement message with zero lifetime to the MAG) However, there are rules for removing an entry in the LMA.

Fig 5 shows some examples of the Binding Cache Entry.

Case 1

State	Home Prefix	CoA	MN-ID	IF-ID	Tunnel-IF-ID	Access Type	Multi-Homing Mode
Idle Idle	HNP1 HNP2	MAG1.PCoA MAG2.PCoA	NAI NAI	IF-ID IF-ID	T1-ID T2-ID	WLAN WiMAX	ON

Case 2

State	Home Prefix	CoA	MN-ID	IF-ID	Tunnel-IF-ID	Access Type	Multi-Homing Mode
Idle Active	HNP1 HNP2	MAG1.PCoA MAG2.PCoA	NAI NAI	IF-ID IF-ID	T1-ID T2-ID	WLAN WiMAX	ON

Case 3

State	Home Prefix	CoA	MN-ID	IF-ID	Tunnel-IF-ID	Access Type	Multi-Homing Mode
Active Active	HNP1 HNP2	MAG1.PCoA MAG2.PCoA	NAI NAI	IF-ID IF-ID	T1-ID T2-ID	WLAN WiMAX	ON

Case 4

State	Home Prefix	CoA	MN-ID	IF-ID	Tunnel-IF-ID	Access Type	Multi-Homing Mode
Active Active	HNP1 HNP2	MAG2.PCoA MAG2.PCoA	NAI NAI	IF-ID IF-ID	T2-ID T2-ID	WiMAX WiMAX	OFF

Case 5

State	Home Prefix	CoA	MN-ID	IF-ID	Tunnel-IF-ID	Access Type	Multi-Homing Mode
Idle Idle	HNP1 HNP2	MAG2.PCoA MAG2.PCoA	NAI NAI	IF-ID IF-ID	T2-ID T2-ID	WiMAX WiMAX	OFF

Case 6

State	Home Prefix	CoA	MN-ID	IF-ID	Tunnel-IF-ID	Access Type	Multi-Homing Mode
Idle Active	HNP1 HNP2	MAG2.PCoA MAG2.PCoA	NAI NAI	IF-ID IF-ID	T2-ID T2-ID	WiMAX WiMAX	OFF

Fig. 5. Examples of Binding Cache Entries

In Cases 1 and 2, the LMA cannot remove any Home Network Prefixes before receiving the Deregistration Proxy Binding Update from the MAG even though the idle time has expired. This is because there would be no available entry for the multi-homed MN when it tried to send some flows again if the LMA had removed the HNP1 in the Binding Cache Entry. Therefore, the LMA should keep the Binding Cache Entry without any modification.

In Cases 3 and 4, the LMA should not remove any Home Network Prefixes. Especially in case 4, when the multi-homed MN is reattached to the WLAN or attached to a new WLAN, a new Home Network Prefix should be allocated to the reattached interface or newly attached interface by the LMA.

In Cases 5 and 6, the LMA removes the HNP1, and then the HNP1 can be reused when the multi-homed MN is reattached to the WLAN or attached to a new WLAN.

3.3 Mobile Access Gateway Consideration

A Mobile Access Gateway (MAG) processes a Mobility Session Migration message to send the Home Network Prefix which is assigned to the detached interface of a multi-homed MN to the attached interface of the multi-homed MN. For instance, when a MAG receives a Mobility Session Migration message from an LMA, it can send a Router Advertisement message with multiple home network prefix options. The home network prefix options will include more than one home network prefix. The first home network prefix option includes the Home Network Prefix that is assigned for the given active interface of the multi-homed MN. The other home network prefix options include home network prefixes that were assigned for the given detached interface of the multi-homed MN.

4 Operations of Proposed Multi-homing Support with PMIPv6

In this section, we propose the operations required for multi-homing support of PMIPv6 based on the considerations in Section 3. Fig 6 shows the signaling call flow when a multi-homed MN enters a Proxy Mobile IPv6 domain. This call flow is the same as that in the current Proxy Mobile IPv6 specification. The difference between the current Proxy Mobile IPv6 and the proposed scheme is that the binding cache entry of the LMA has a 1-bit field that we call the multi-homing mode field to distinguish whether a mobile node is a multi-homed mobile node or not. In addition, we define a 1-bit state field in the Binding Cache Entry to determine for each entry whether the entry is in the active state or not. For example, when the LMA receives Proxy Binding Update messages from MAG1 and MAG2 respectively, the LMA checks its Binding Cache Entry. If the LMA finds matching entries that have the same MN-ID but have different tunnel-IF-IDs, the Multi-Homing Mode field is set to 1 by the LMA. In this way, the LMA can manage each mobility session for the multi-homed MN under the same binding cache entry.

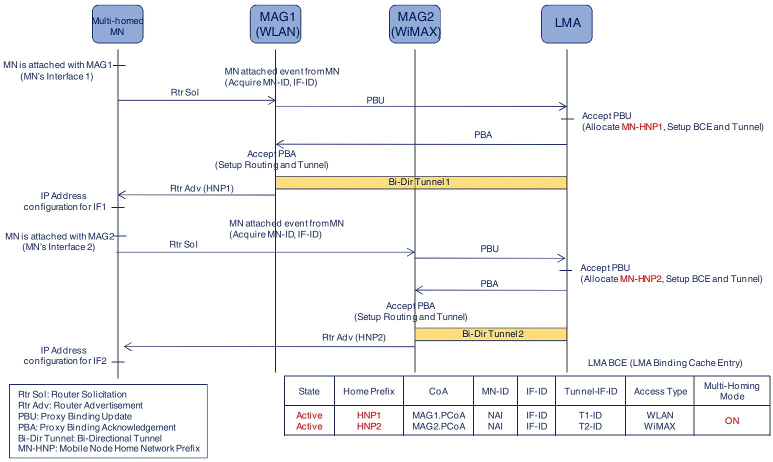


Fig. 6. Signaling call flow for multi-homed Mobile Node attachment

Figure 7 shows the signaling call flow of inter-technology handover. The operations of Inter Access Technology Handover are as follows.

- Operations

(1) CN sends packets to MN through bi-directional tunnel 1 between LMA and MAG1

(2) MN is detached from MAG1

(3) MAG 1 sends a De-Registration Proxy Binding Update message to LMA

(4) LMA accepts the De-Registration Proxy Binding Update message and starts buffering packets for MN

(5) LMA sends a Mobility Session Migration message that includes the home network prefix (HNP 1) that was assigned for MN's Interface 1 to MAG2.

(6) MAG2 sends a Mobility Session Migration Acknowledgement message to LMA as a response to the Mobility Session Migration.

(7) MAG2 adds the binding of MN to its binding lists

(8) LMA sends a Proxy Binding Acknowledgement message to MAG1 as a response to the De-Registration Proxy Binding Update.

(9) MAG1 removes the binding of MN from its binding lists

(10) LMA updates the Binding Cache Entry

(11) MAG2 sends a Router Advertisement message with multiple prefix option (HNP1, HNP2) to MN

(12) Multi-homed node generates a global IPv6 address using received home network prefix (HNP1) and then assigns the IPv6 address to MN's Interface 2. (As for HNP2, the lifetime is only updated to use it on interface 2)

(13) LMA sends an ICMPv6 Echo Request to MN. In this message, the Source Address field is set to the Local Mobility Anchor Address and the Destination Address field is set to a combination of the home network prefix(HNP1) and IF-ID.

(14) LMA receives an Echo Reply from MN.

(15) LMA starts to forward the packets to MN

(16) Therefore, CN can continually communicate with MN.

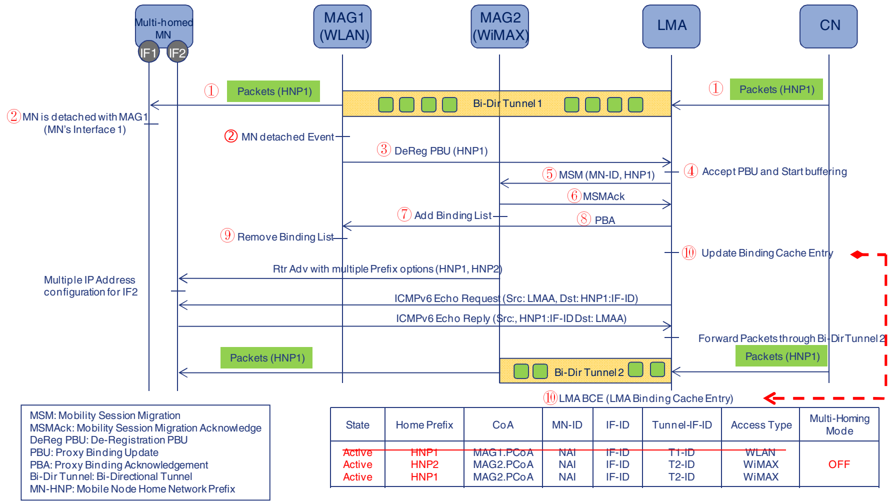


Fig. 7. Signaling call flow for inter-technology handover from overlay network to single network

Fig 8 shows the signaling call flow of inter-technology handover when the MN is reattached to the previous MAG1. The difference between the cases in Fig 7 and Fig 8 is that both HNP1 and HNP2 are in the active state. This means that the LMA still sends packets to HNP1 and HNP2. In that situation, a New Home Network Prefix should be allocated for the reattached Interface.

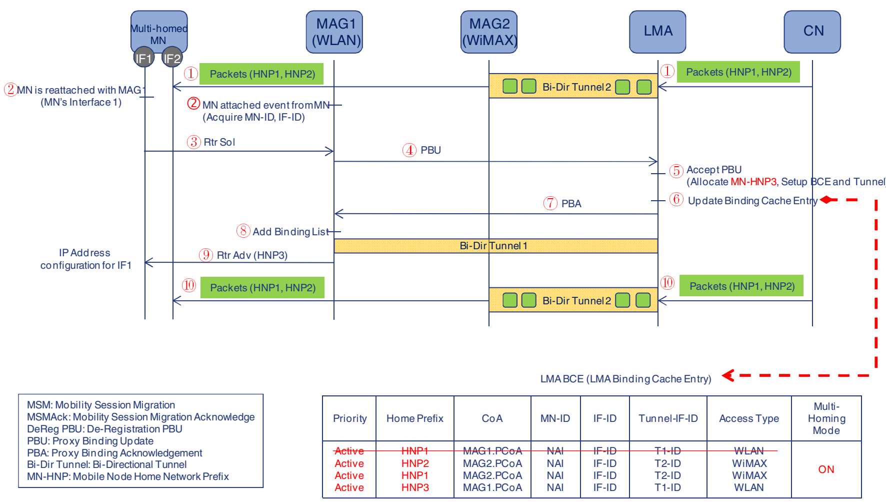


Fig. 8. Signaling Call Flow for inter-technology handover from single network to overlay network

5 Performance Evaluations and Results

In order to evaluate the performance of the proposed scheme, we created a simulation network topology using QualNet 4.5[13], as shown in Fig 9. The main purpose of this simulation is to demonstrate application session continuity when a multi-homed MN is moved across different networks.

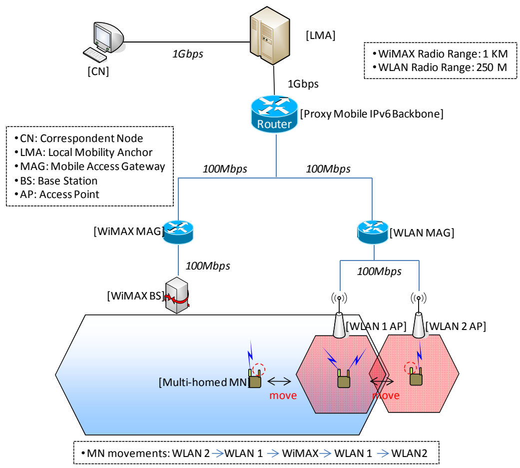


Fig. 9. Network topology over WiMAX and WLAN interworking systems

Based on the network topology, we configured a scenario which includes four cases, with the MN's moving at speeds of 5 km/h, 20 km/h, 60 km/h and 100 km/h. In the simulation, the multi-homed MN starts in WLAN 2 and a VoIP (UDP) session between the multi-homed MN and CN is established. Then it moves into the WLAN 1 coverage within the WiMAX coverage. At this time an FTP (TCP) session between the multi-homed MN and CN is established through the WiMAX. After that it moves out of the WLAN 1 and is solely under WiMAX coverage. At this time, the VoIP session is moved from WLAN 1 to WiMAX. Finally, the multi-homed MN goes back to the WLAN 2 again. At this time, the VoIP and FTP sessions are moved from WiMAX to WLAN 2 according to the proposed PMIPv6 scheme. The simulation parameters are given in Table 2.

Table 2. Simulation parameters

Parameters	Values
FTP traffic	1024 bytes (CN continuously sends MN items of 1024 bytes each after MN is attached to WiMAX until the end of the simulation.)
VoIP traffic	512 bytes (The inter-departure time for each item is 1 second.)
MN moving speed	5, 20, 60, 100km/h
WLAN data rate	11Mbps
WiMAX data rate	7Mbps (The data rate is achieved with OFDM 512 FFT size and 16QAM 1/2 over 5 MHz channel bandwidth.)
Simulation time	5km/h case(1440 sec), 20km/h case(540 sec), 60km/h case(180 sec), 100km/h case(110 sec)

Fig 10 shows the multi-homed MN's cumulative data for VoIP traffic during the WLAN-WiMAX-WLAN inter-technology handovers. The VoIP application session started at 5 seconds at a rate of 512 bytes/sec. When the multi-homed MN moved out of the WLAN 1 coverage after 181 seconds, the VoIP session moved from the WLAN 1 to WiMAX in the proposed PMIPv6 case. However, in the standard PMIPv6 case as shown in Fig 10, the VoIP session stopped after the multi-homed MN was detached to the WLAN 1. Since, the multi-homed MN was detached to the WLAN 1, the WLAN MAG sent a deregistration proxy binding update message to the LMA thus, the LMA removed the mobility entry for the VoIP from its BCE in the standard PMIPv6. On the other hand, when the multi-homed MN was detached to the WLAN 1, the HNP that was assigned to the WLAN interface of the multi-homed MN was reassigned to the WiMAX interface of the multi-homed MN according to the proposed PMIPv6 scheme so that the VoIP session did not stop until the end of the simulation. Finally, the multi-homed MN was detached to the WiMAX and reattached to the WLAN2 at 452 seconds. At this time, the VoIP session moved from WiMAX to WLAN 2.

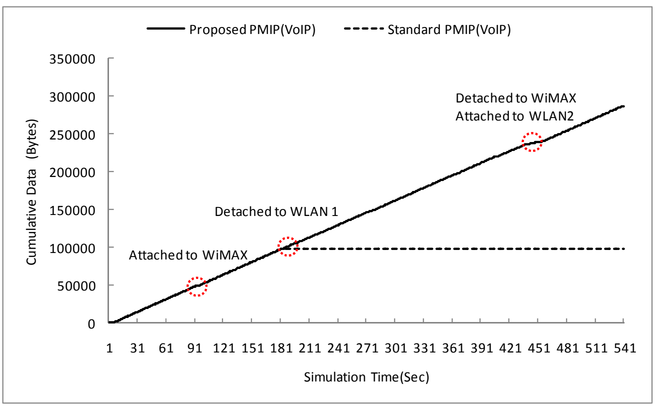


Fig. 10. MN's cumulative data for VoIP traffic(MN moving speed=20km/h)

Fig 11 shows the multi-homed MN's cumulative data rate for FTP traffic during the WiMAX-WLAN inter-technology handover. The FTP session started at 150 seconds after the multi-homed MN was attached to WiMAX at 107 seconds. The multi homed MN moved towards the WiMAX Base Station between 107 seconds and 270 seconds and far from the WiMAX Base Station between 271 seconds and 452 seconds. During those time periods, we confirmed that the data rate at each second was very low. The reason is that as the multi-homed MN moves far from the WiMAX Base Station, the path loss increases and the signal power at the WiMAX Base Station decreases, thus the uplink and downlink data rate at the edge of WiMAX cell falls to a low rate. On the other hand, the data rate was increased dramatically between 452 seconds and 540 seconds due to the WLAN's high bandwidth.

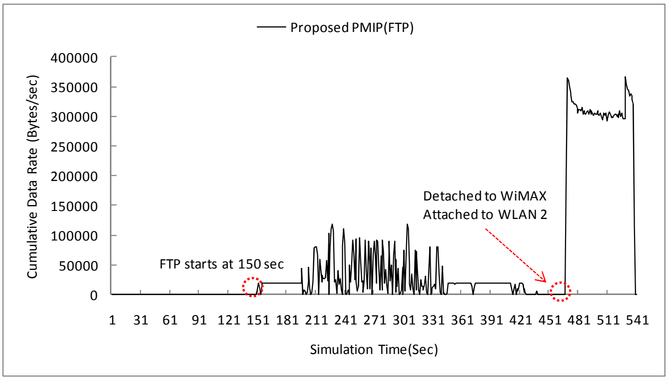


Fig. 11. MN's cumulative data rate for FTP traffic(MN moving speed=20km/h)

In this manner, we have evaluated our proposed scheme with different MN's moving speeds as shown in Fig 12 and Fig 13. Fig 12 shows the cumulative data for VoIP and FTP at different MN's moving speeds and Fig 13 depicts the multi-homed MN's average throughput for FTP and VoIP. As we explained in this section, each traffic is always transmitted through at least one of the interfaces of multi-homed MN so that the proposed scheme shows better cumulative data and average throughputs than the standard scheme. It means that the proposed scheme is expected to be an appropriate scheme rather than the standard scheme for such a heterogeneous network.

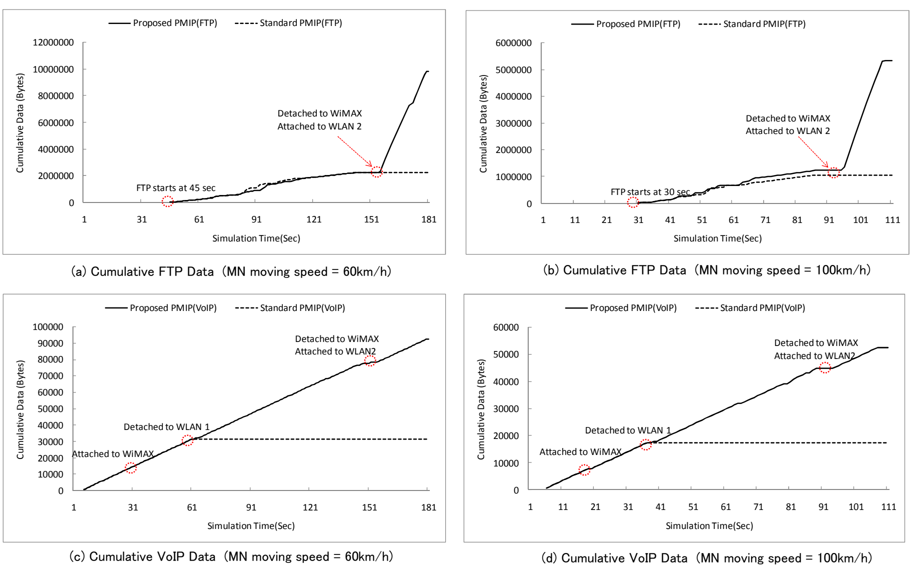
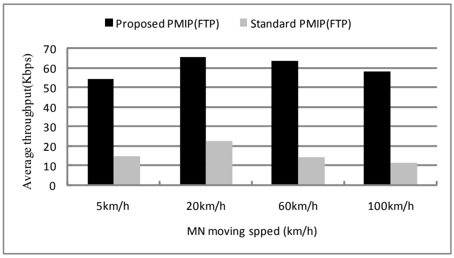
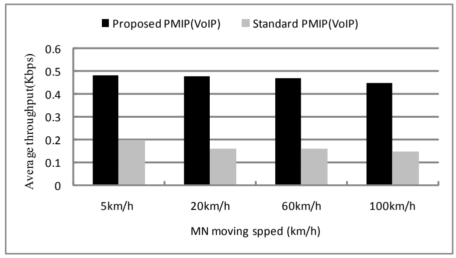


Fig. 12. Cumulative data for VoIP and FTP



(a) Average FTP traffic throughput



(b) Average VoIP traffic throughput

Fig. 13. MN's average throughput for FTP and VoIP

Therefore, the simulation results show that application session continuity is guaranteed with our proposed PMIPv6 scheme.

6 Conclusion

This paper proposed an enhanced multi-homing support scheme for PMIPv6. With the proposed scheme, a multi-homed MN is able to assign multiple home-network prefixes to its interface so that it can keep application sessions continuously during a handover between interfaces. To achieve this, we defined the MSM/MSMA messages required for the LMA to allocate the HNP that was assigned to the detached interface of a multi-homed MN to the currently attached interface. By doing so, application session continuity is guaranteed while the multi-homed MN is moved across different networks. Simulation results confirmed that the FTP and VoIP traffic was transmitted continuously from the detached interface to the attached interface over the WiMAX and WLAN interworking heterogeneous networks.

References

1. Liu, X., Xiao, L.: A Survey of Multihoming Technology in Stub Networks: Current Research and Open Issues. IEEE Network 21(3), 32–40 (2007)
2. Nordmark, E., Bagnulo, M.: Shim6: Level 3 Multihoming Shim Protocol for IPv6, IETF RFC 5533 (June 2009)
3. Gundavelli, S., et al.: Proxy Mobile IPv6, RFC 5213 (August 2008)
4. 3GPP, Proxy Mobile IPv6 (PMIPv6) based Mobility and Tunnelling protocols; Stage 3 (Release 10), TS 29.275 V10.0.0 (2010)
5. WiMAX Forum, `http://www.wimaxforum.org/`
6. Mitra, S.: Seamless Mobility Management and QoS Support for Multihomed Mobile Node in Heterogeneous Wireless Networks. In: ICIIS 2010, pp. 145–150 (July 2010)
7. Thomson, S., et al.: IPv6 Stateless Address Autoconfiguration. IETF RFC 4662 (September 2007)
8. IEEE, Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority, `http://standards.ieee.org/db/oui/tutorials/EUI64.html`
9. Narten, T., et al.: Privacy Extensions for Stateless Address Autoconfiguration in IPv6, IETF RFC 3041 (January 2001)

10. Johnson, D., Perkins, C., Arkko, J.: Mobility Support in IPv6, RFC 3775 (June 2004)
11. Patel, A., et al.: Mobile Node Identifier Option for Mobile IPv6 (MIPv6), IETF RFC 4283 (November 2005)
12. Aboba, B., Beadles, M., Arkko, J., Eronen, P.: The Network Access Identifier, RFC 4282 (December 2005)
13. Qualnet simulator, `http://www.scalable-networks.com/`

Fast Handover Scheme Based on Mobility Management of Head MAG in PMIPv6

NamYeong Kwon, Hongsuk Kim, Seung-Tak Oh, and Hyunseung Choo

School of Information and Communication Engineering
Sungkyunkwan University
300, Chunchun-dong, Jangan-gu, Suwon, Kyunggi-do, 440-746, Korea
{nykwon,hassium84}@skku.edu, {xrossat,choo}@ece.skku.edu

Abstract. Proxy Mobile IPv6 (PMIPv6) is a network-based mobility management protocol that supports network mobility, regardless of whether or not a Mobile Node (MN) supports mobility protocol, such as Mobile IPv6 (MIPv6) and Hierarchical MIPv6 (HMIPv6). However, PMIPv6 protocol does not consider the packet loss during MN's handover. We propose a fast handover scheme using the Head Mobile Access Gateway (MAG) to provide a fast and seamless handover service in PMIPv6. The proposed scheme does not violate the principle of PMIPv6 by maintaining mobility information of the MN in the Head MAG. The Head MAG is granted to an MAG deployed at an optimal location in a Local Mobility Anchor (LMA) domain. The Head MAG reduces packet loss and enables fast handover by providing mobility information of the MN to the MAGs in the same LMA domain. In this paper, we analyze an analytic performance evaluation in terms of signaling cost and size of packet loss. The analytic performance evaluation shows that the proposed scheme is compared with basic PMIPv6 and the previous proposed scheme using the Neighbor Detection (ND) message. The proposed scheme reduces signaling cost by over 30% compared to the previous proposed scheme using ND message and packet loss by over 78% compared to basic the PMIPv6 scheme.

Keywords: Proxy Mobile IPv6, PMIPv6, Head MAG, Fast handover, Seamless handover, Buffering, Mobility information.

1 Introduction

Recent IT technologies are rapidly developing in portable device environments supporting networks. The number of users who want to get internet services keeps increasing. Research provisioning users with reliable Quality of Service (QoS) in mobile ubiquitous environments is abundant in the literature [1].

Proxy Mobile IPv6 (PMIPv6) is a next generation technology able to satisfy user requests. It is a network based mobility protocol suggested by the Internet Engineering Task Force Network-based Localized Mobility Management Working Group [2]. In PMIPv6, a Mobile Node (MN) is not involved in handover signaling; mobility can be provided without any MN's handover signaling. PMIPv6

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 181–192, 2011.

performs better than MIPv6 in many aspects (e.g., fewer resources in wireless, less energy consumption of an MN, and lower handover signaling cost) due to network based mobility protocol. The handover latency is normally decreased in PMIPv6 networks, because the MN's IP address does not need to be changed, regardless of local handovers.

However, packet losses caused during handover disruption are not guaranteed. This problem incurs low reliability. Therefore, many researches have focused on the mobility issues of a handover. A handover scheme using L2 signaling [3], a fast handover scheme with a new entity [4], and a handover scheme with information delivery between neighboring MAGs have been proposed to solve this problem [5][6]. Most solutions are either applied in limited network environments or use an additional entity of new functionality and high cost signaling, such as multicast. These schemes are easily adapted to PMIPv6 protocol with low signaling for general purposes.

In this paper, we propose a seamless fast handover scheme that provides high quality mobility service. The proposed scheme introduces a Head MAG that manages MN's mobility information during handover. This scheme reduces signaling cost because it does not use L2 messages or additional entities as in the previous schemes. Thus, it can be used widely.

The remainder of the paper is structured as follows. Section 2 outlines the PMIPv6, signaling in handover, mobility management technique, and problems of previously proposed fast handover scheme using the ND message [6]. Section 3 depicts the proposed scheme of fast handover using the Head MAG. Section 4 and 5 will compare the proposed scheme to the previous proposed scheme and show the performance evaluation of signaling cost in handover and amount of packet loss. We conclude the paper in section 6.

2 Related Work

The network based mobility protocol, PMIPv6, is suggested to solve problems of the host based mobility management protocol, MIPv6. As stated above, the outstanding feature of PMIPv6 protocol is that the MN does not need to participate in handover signaling, which is performed between the MAG and LMA during handovers. Therefore, only IPv6 stack is sufficient for the mobility in PMIPv6. In this section, we first describe the basic architecture and handover procedure of PMIPv6, and then introduce the previous fast handover scheme using the ND message.

2.1 Proxy Mobile IPv6

The most important feature of the PMIPv6 is network based mobility protocol. That is, the MAG and the LMA are in charge of the mobility of an MN in PMIPv6 domain. On the other hand, the MN in the host-based mobility protocol, MIPv6, roles the main function of mobility.Therefore the MNs with an IPv6 protocol stack are able to get service with mobility in PMIPv6, since the PMIPv6 protocol does not require any mobility function of MN.

PMIPv6 introduces the MAG and LMA that manage the mobility of the MN. When the MN moves from the previous MAG (pMAG) to the new MAG (nMAG), it sends its ID to the nMAG in the L2 handover process. The nMAG authenticates this information via an Authentication, Authorization, and Accounting (AAA) server. Next, the nMAG acquires the MN's profiles for handover, such as Home Network Prefix (HNP), LMA address, and address setting method from the AAA server.

After the authentication, the nMAG sends the Proxy Binding Update (PBU) message to the LMA for location registration of the MN. The PBU message contains the ID of the MN and the HNP. The LMA can get information of the MN from it. The LMA checks if information of the MN is in the Binding Cache Entry (BCE). If not, it adds information of the MN to the BCE to provide the service. When the authentication process between the AAA server and the LMA finishes normally, the LMA transmits the Proxy Binding Acknowledgement (PBA) message to the nMAG to create a tunnel between the LMA and nMAG. After the tunnel is created, the packets, which are sent to the MN from the CN, are transmitted to the MN via the LMA and nMAG. However, some of these packets are inevitably lost during handover delay. This delay is caused by the authentication process and tunnel establishment in the handover process.

2.2 Fast Handover Scheme Using the ND message

This scheme is proposed to provide seamless service during an MN's handover. In this scheme, the pMAG provides the MN-profile including the MN-HNP, MN-ID, and LMA address to 1 hop-distance neighbor MAGs. The pMAG recognizes the Link Going Down (LGD) trigger, as soon as the disconnection between the pMAG and the MN occurs. If the pMAG detects the LGD triggering, it prevents packet loss by passing the message containing the MN-profile to all its 1 hop-distance neighbor MAGs. While the LGD trigger occurs, all packets related to the MN store in the pMAG's buffer to provide seamless handover service in the proposed scheme. We simply name this the Neighbor MAGs scheme.

In this scheme, when the LGD triggering occurs from the pMAG, the pMAG delivers the Neighbor Detection (ND) messages to all capable MAGs within 1 hop-distance and starts buffering all the packets destined to the MN. The ND message contains the MN-profile. Then, the pMAG informs the LMA of the disconnection status of the MN with the DeReq message. The LMA, which gets the DeReq PBU message, starts buffering the packets destined to the MN and transmits the PBA message to the pMAG.

When the MN handovers to the nMAG, the nMAG sends the Handover Notification (HN) message to the pMAG, as a response to the ND message from the pMAG. pMAG, which receives the HN message, starts forwarding the buffered packets to the nMAG. Next, the LMA starts forwarding the buffered packets to the nMAG, as soon as a tunnel between the nMAG and LMA is established. This scheme is able to prevent more packet losses than the basic PMIPv6, since the MN can detect handover in advance and performs buffering. However, it incurs significant load due to buffering from both LMA and MAG. It also causes unnecessary signaling when the MN performs a handover to another domain, because the ND message is sent before the MN handover.

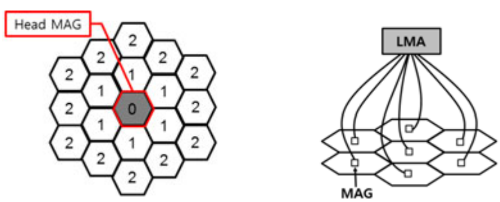
3 Proposed Scheme

We propose a scheme using a Head MAG to obtain mobility information of an MN, complying with the principles of PMIPv6 strictly. The Head MAG is one of MAGs in an LMA domain. An MAG, which is located in the optimal position to perform efficient communication within a PMIPv6 domain, is selected as the Head MAG. It manages the mobility information of the MN, in addition to the original role of the MAG.

3.1 Definition of a Head MAG

The proposed scheme using the Head MAG provides seamless service when the MN handovers. When an MN moves from a pMAG to an nMAG, the nMAG requests the information of the pMAG to the Head MAG in the same LMA domain. There is no unnecessary signaling from inter-domain handover in our proposed scheme, because the pMAG does not request information of the MN to the Head MAG, when the MN's inter-domain handovers.

There is one Head MAG in an LMA domain. That is, the Head MAG is in the middle of the LMA domain and selected by the network manager. Fig. 1(a) shows the position of the Head MAG; all MAGs are expressed in a hexagonal model. Fig. 1(b) shows the location of the LMA. Hexagons denote MAGs. An LMA domain consists of multiple MAGs. The Head MAG is in charge of managing the MN's mobility information within its LMA domain. The Head MAG is located at the position that has the shortest distance among neighbor MAGs, when intra-MAG communication occurs.



(a) (b)

Fig. 1. The location of the Head MAG and the LMA: (a) the location of the Head MAG, (b) the location of the LMA

The Head MAG maintains the addresses of all MAGs in the same LMA domain and MN-ID. When an MN moves to an nMAG, the nMAG obtains the address of pMAG from the Head MAG; the nMAG, to which the MN is attached, becomes a serving MAG. The Head MAG is able to maintain the information of the MAG and MN. At this time, a tunnel is constructed between the nMAG and the pMAG, using the address of pMAG, to transfer buffering packets.

An LMA maintains the location information of an MN in the original. Therefore, there could be significant loads, because the LMA handles all the signaling related to MAGs and the location of the LMA may not be optimal to communicate with MAGs. However LMA has no additional load in the proposed scheme using the Head MAG. The mobility management of the MN can be controlled more effectively due to the optimal location of the Head MAG. The Head MAG also causes a small load due to the messages when the MN's handover.

3.2 Handover Procedure

There is an additional signaling message in the proposed scheme due to the existence of the Head MAG for fast handover. The nMAG uses a HI (Handover Initiate) and a HAck (Handover Acknowledgement) message to obtain address information of the pMAG from the Head MAG. The signaling of the proposed scheme is divided into two ways (i.e., the signaling when an MN first attaches to the LMA domain and when a handover occurs within the same LMA domain).

We introduce the signaling procedure when an MN first attaches in the LMA domain. The MAG to which an MN attached detects movement of the MN, when the MN enters to the LMA domain for the first time. The MAG sends an HI message, which contains MN-ID information to the Head MAG to obtain the previous MAG address. The MAG also communicates with an AAA server to get the MN identification simultaneously. The Head MAG checks the attachment information of the MN with its MN-ID table. However, the Head MAG does not have the address information of the pMAG related to the MN, because the MN attaches to the LMA at first. Therefore, the Head MAG sends a HAck message to the MAG without address information of the pMAG. Next, the table in the Head MAG saves the preset MN-ID and MAG address. The MAG recognizes that the MN attaches to its LMA domain first, and hence the MAG does not send an additional request message to the Head MAG. The signaling procedure is the same as the general PMIPv6 except this signaling procedure.

The proposed scheme provides mobility information of the MN that attaches to one of the MAGs in the same domain. Therefore, seamless handover service can be provided within the same domain. Fig. 2 shows signaling when an MN performs a handover in the same LMA domain. The MN disconnects to the pMAG and connects to the nMAG due to the movement of the MN. The pMAG starts to store the data packets in buffer, which are going to the MN, after disconnecting with the MN. The nMAG detects the attachment of the MN and requests the address of the pMAG to the Head MAG using an HI message. The nMAG also communicates with the AAA server to authenticate the MN. Fig. 2 abridges the authentication of the MN with the AAA server.

The Head MAG, which gets the HI message, checks the address of pMAG, and sends the HAck message with the pMAG to the nMAG. After getting the HAck message, the nMAG sends a forwarding request message to the pMAG to obtain buffering packets. A tunnel is constructed between the pMAG and the nMAG. The pMAG starts to forward the buffering packet via the tunnel. The buffering packets are delivered from the pMAG to the nMAG via the tunnel,

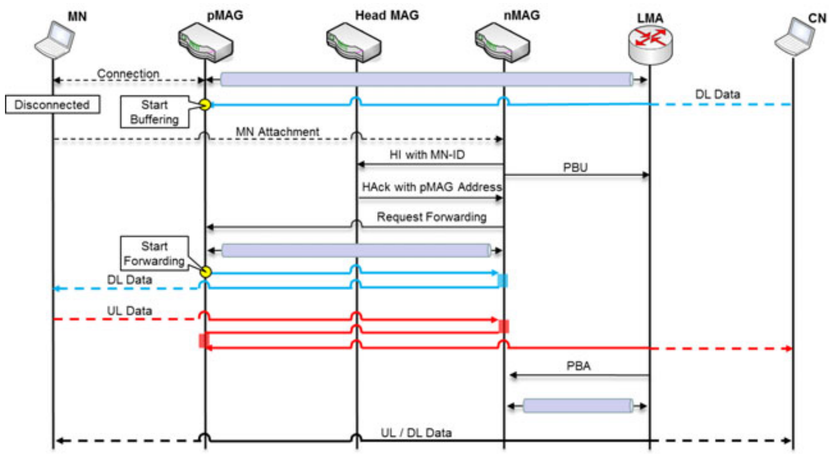


Fig. 2. Message Flow for the Handover

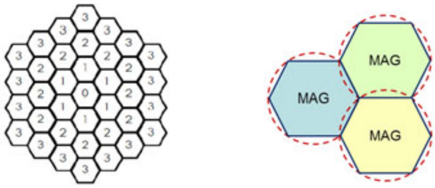
and then from the nMAG to the MN. The MN sends packets to the nMAG and the nMAG sends the packets to the pMAG via the tunnel, when the packets are transmitted from the MN to the CN. The pMAG sends packets to the LMA via the original tunnel and the packets are finally delivered to the CN. After all the procedures to establish a tunnel between the nMAG and the LMA are completed, the tunnel can be used between the nMAG and the LMA. Later, packets from the uplink and downlink are sent to the nMAG directly, without the pMAG.

4 Performance Model

In this section, we evaluate the performance against PMIPv6, using the information of neighbor MAGs, and the proposed scheme. First, we describe our analytical model, including the network and traffic models. Second, we evaluate the performance using the analytical modeling.

4.1 Network and Mobility Model

We use a hexagonal network model for performance evaluation [7]. A layered hexagonal network model, as shown in Fig. 3(a), is used as our network model. Each cell denotes an MAG and the cells construct an LMA domain. Fig. 3(b) illustrates the circular shape coverage of an MAG. A number in the cell denotes the level of the cell. We calculate the total number of MAGs in the LMA domain. We use the Fluid-flow network mobility model, a well-known mobility model in the mobile network field. The mobility model represents the characters of an MN. The mobility model considers the velocity and direction of an MN. We calculate the cell-crossing rate of an MN intra-domain and inter-domain. In addition, we



(a) (b)

Fig. 3. Network Model: (a) hexagonal network model, (b) circular shape coverage of an MAG

calculate the average number of movements using the number of MAGs and session arrival rate of an MN.

Let μ_c be the cell-crossing rate. μ_c can be calculated, as shown in Equation(1) [7]. ν is the average velocity of the MN, R is the coverage radius of an MAG, S is the extent of a cell. μ_d is the inter-domain cell-crossing rate of the MN. μ_d can be calculated, as shown in Equation (3). N is the number of MAGs in an LMA domain. μ_s is the intra-domain cell-crossing rate of the MN. μ_s can be calculated, as shown in Equation (4).

$$\mu_c = \frac{2 \cdot \nu}{\sqrt{\pi \cdot S}} = \frac{2 \cdot \nu}{\pi \cdot R} \tag{1}$$

$$S = \pi \cdot R^2 \tag{2}$$

$$\mu_d = \frac{\mu_c}{\sqrt{N}} \tag{3}$$

$$\mu_s = \mu_c - \mu_d = \mu_c \cdot \frac{(\sqrt{N} - 1)}{\sqrt{N}} \tag{4}$$

The mobility model can calculate the average number of movements, $E[N_c]$. $E[N_d]$ is the average number of movements in the inter-domain. $E[N_s]$ is the average number of movements in the intra-domain[7]. $E[N_c]$, $E[N_d]$, and $E[N_s]$ can be calculated, as shown in Equation (5), (6), and (7).

$$E[N_c] = \frac{\mu_c}{\lambda s} \tag{5}$$

$$E[N_d] = \frac{\mu_d}{\lambda s} \tag{6}$$

$$E[N_s] = \frac{\mu_s}{\lambda s} = \frac{(\mu_c - \mu_d)}{\lambda_s} \tag{7}$$

4.2 Analytical Modeling

We evaluate performance using the analytical model. In this section we compare the signaling cost and packet loss when the MN's handover.

4.2.1 Signaling Cost

In basic PMIPv6, when the MN handovers to another MAG, the signaling cost is the sum of the location update and authentication between an MAG and an LMA. The signaling cost of authentication with the AAA server occurs in all schemes. Therefore, we do not consider the authentication signaling cost. SC_{PMIPv6} is the total signaling cost in the basic PMIPv6. SC_{PMIPv6} can be calculated, as shown in Equation (8) [8][9].

$$SC_{PMIPv6} = E[N_c] \cdot (SC_{PBU} + SC_{PBA}) \tag{8}$$

The PBU and the PBA signaling always occur in PMIPv6, when the MN handovers to another MAG. SC_{PBU} and SC_{PBA} are the signaling cost of PBU and the signaling cost of PBA, respectively. SC_{PBU} and SC_{PBA} can be calculated, as shown in Equation (9), (10). α is the transmission unit cost in a wired link, $d_{MAG-LMA}$ is the number of hops between an MAG and an LMA, L_{PBU} is the size of the PBU message. PC_R is the processing cost in the router, PC_{LMA} is the processing cost in an LMA. L_{PBA} and PC_{MAG} are the size of the PBA message and the processing cost in the MAG, respectively.[8][9][10]

$$SC_{PBU} = \alpha \cdot (d_{MAG-LMA} \cdot L_{PBU}) + (d_{MAG-LMA} - 1) \cdot PC_R + PC_{LMA} \tag{9}$$

$$SC_{PBA} = \alpha \cdot (d_{LMA-MAG} \cdot L_{PBA}) + (d_{LMA-MAG} - 1) \cdot PC_R + PC_{MAG} \tag{10}$$

The signaling cost of the scheme using the movement information of the MNs to the neighbor MAGs can be calculated, as shown in Equation (11). SC_{ND} and SC_{HN} are the signaling cost of the neighbor detection message and the signaling cost of the handover notification message, respectively. SC_{ND} and SC_{HN} can be calculated, as shown in Equation (12), (13).

$$SC_{NeighborMAG} = E[N_c] \cdot (SC_{PBU} + SC_{PBA}) + E[N_c] \cdot (6 \cdot SC_{ND} + SC_{HN}) \tag{11}$$

$$SC_{ND} = \alpha \cdot L_{ND} + PC_{MAG} \tag{12}$$

$$SC_{HN} = \alpha \cdot L_{HN} + PC_{MAG} \tag{13}$$

The proposed scheme adds the Hi, HAck, and RF messages. The signaling cost of the proposed scheme can be calculated, as shown in Equation (16). $SC_{HeadMAG-Intra}$ and $SC_{HeadMAG-Inter}$ are the signaling cost of the movement in intra-domain and inter-domain, respectively. The scheme does not send the RF message for an MN's inter-domain handover.

$$SC_{HeadMAG-Intra} = SC_{HI-Intra} + SC_{HAck-Intra} + SC_{RF} \tag{14}$$

$$SC_{HeadMAG-Inter} = SC_{HI-Inter} + SC_{HAck-Inter} \tag{15}$$

$$SC_{HeadMAG} = E[N_c] \cdot ((9) + (10)) + E[N_s] \cdot (14) + E[N_d] \cdot (15) \tag{16}$$

Therefore, we calculate the intra-domain and inter-domain cost. $SC_{HeadMAG-Intra}$ and $SC_{HeadMAG-Inter}$ can be calculated, as shown in Equation (17)-(21). L is the level of MAG in an LMA domain. That is, L value is the maximum hop distance between an MAG and the HeadMAG.

$$SC_{HI-Intra} = \alpha \cdot (d_{HeadMAG} \cdot L_{HI}) + (d_{HeadMAG} - 1) \cdot PC_R + PC_{HeadMAG} \tag{17}$$

$$SC_{HAck-Intra} = \alpha \cdot (d_{HeadMAG} \cdot L_{HAck}) + (d_{HeadMAG} - 1) \cdot PC_R + PC_{MAG} \quad (18)$$

$$SC_{HI-Inter} = \alpha \cdot (L \cdot L_{HI}) + (L-1) \cdot PC_R + PC_{HeadMAG} \quad (19)$$

$$SC_{HAck-Inter} = \alpha \cdot (L \cdot L_{HAck}) + (L-1) \cdot PC_R + PC_{MAG} \quad (20)$$

$$SC_{RF} = \alpha \cdot (d_{HeadMAG} \cdot L_{RF}) + (d_{HeadMAG} - 1) \cdot PC_R + PC_{HeadMAG} \quad (21)$$

$d_{HeadMAG}$ is the average number of hops between the Head MAG and other MAGs. $d_{HeadMAG}$ can be calculated, as shown in Equation (22).

$$d_{HeadMAG} = \frac{\sum_{c=1}^{L} 6c^2}{\sum_{c=1}^{L} 6c + 1} \quad (22)$$

4.2.2 Packet Loss

The basic PMIPv6 does not consider the packet loss in performance evaluation and the scheme cannot prevent the physical packet loss. Accordingly packets are lost when the MNs are detached from the pMAG even after establishing a tunnel between the nMAG and the LMA. $S_{Drop-PMIPv6}$ is the size of packet loss in the basic PMIPv6. D_{PMIPv6} is is handover delay of MN's handover. S_{PMIPv6} can be calculated, as shown in Equation (23)-(28) [8][9][10]. t_{mr} is the transmission delay between MN and AP, t_{ra} is transmission delay between AP and MAG. t_{mr} and t_{ta} are the L2 handover delay. t_a is transmission delay between the AAA server and an MAG or an LMA. D_{PBU} and D_{PBA} are transmission delay of PBU and PBA, respectively. B_w is bandwidth of the wired link and λ_p is packet arrival rate in the network.

$$S_{Drop-PMIPv6} = E[N_c] \cdot (\lambda_p \cdot D_{PMIPv6}) \quad (23)$$

$$D_{PMIPv6} = D_{L2} + D_{AAA} + D_{PBU} + D_{PBA} \quad (24)$$

$$D_{L2} = t_{mr} + t_{ra} \quad (25)$$

$$D_{AAA} = 2 \cdot 2t_a = 4t_a \quad (26)$$

$$D_{PBU} = \alpha \cdot (\frac{L_{PBU}}{B_w} + P_t) \cdot (d_{MAG-LMA} - 1) + P_{LMA} \quad (27)$$

$$D_{PBA} = \alpha \cdot (\frac{L_{PBA}}{B_w} + P_t) \cdot (d_{LMA-MAG} - 1) + P_{MAG} \quad (28)$$

The packet loss of the Neighbor MAGs scheme can be calculated, as shown in Equation (29). The packet loss of the proposed scheme can be calculated, as shown in Equation (30).

$$S_{Drop-NeighborMAG} = E[N_d] \cdot (\lambda_p \cdot D_{PMIPv6}) \quad (29)$$

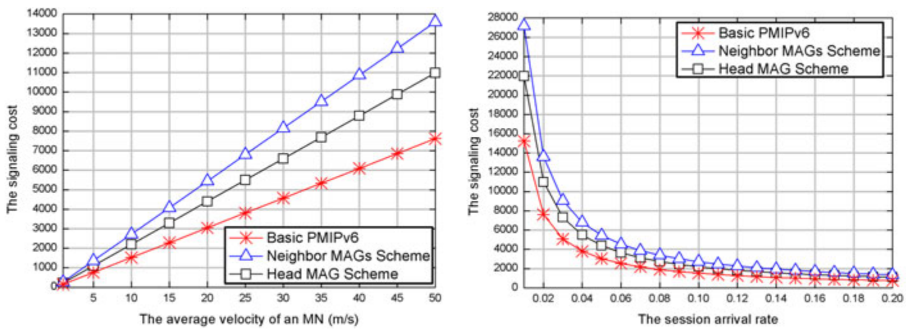
$$S_{Drop-HeadMAG} = E[N_d] \cdot (\lambda_p \cdot D_{PMIPv6}) \quad (30)$$

5 Performance Evaluation

In this section, we estimate the performance of our proposed scheme. We vary the values of parameters and adjust the analytical model [8][9][10].

5.1 Signaling Cost

The signaling cost of the Neighbor MAGs scheme and that of the proposed scheme are the additional message to provide seamless service. Therefore, both schemes incur higher signaling cost compared to the basic PMIPv6. First, we investigate the impact of velocity on the signaling cost. Fig. 4(a) shows the result of signaling cost, when ν is varied from 1 m/s to 50 m/s. In this experiment, we set λ_s to 0.05. As ν increases, the signaling cost also increases because when an MN moves faster, the MN handovers more frequently, thus adding more signaling cost. Fig. 4(b) is the result of signaling cost when λ_s is changed. We vary λ_s from 0.01 to 0.2.



(a) (b)

Fig. 4. Signaling Cost: (a) the impact of the average velocity of an MN, (b) the impact of the session arrival rate

The signaling cost of our proposed scheme increases about 44% compared to the basic PMIPv6. However, this signaling cost is unavoidable to prevent packet loss during the handover of the MN. The signaling cost of Neighbor MAGs scheme is increased by 78% compared to basic PMIPv6. Therefore, our proposed scheme performs better than the Neighbor MAGs scheme.

5.2 Packet Loss

Our proposed scheme aims to reduce packet loss during the MN's handover. The proposed scheme minimizes the packet loss using the buffering on the pMAG and maintains the movement information of the MNs in the Head MAG. Fig. 5(a) shows the size of packet loss when the average velocity of the MN is changed. We set λ_s and λ_p as 0.05 and 5 Mbps, respectively. The value of ν is changed from 1 m/s to 50 m/s. When the average velocity of MN increases, the handover rate of MN increases. Thus, more packets are lost. The proposed scheme decreases packet loss by about 78% compared to basic PMIPv6. Fig. 5(b) shows the size of packets loss when the session arrival rate is changed. We vary λ_s from 0.01 to 0.2. We set ν at 20m/s, and λ_p at 5Mbps. Fig. 5(c) shows the size of packet loss when the packet arrival rate is changed. We vary λ_p from 1 to 10Mbps, ν is 20

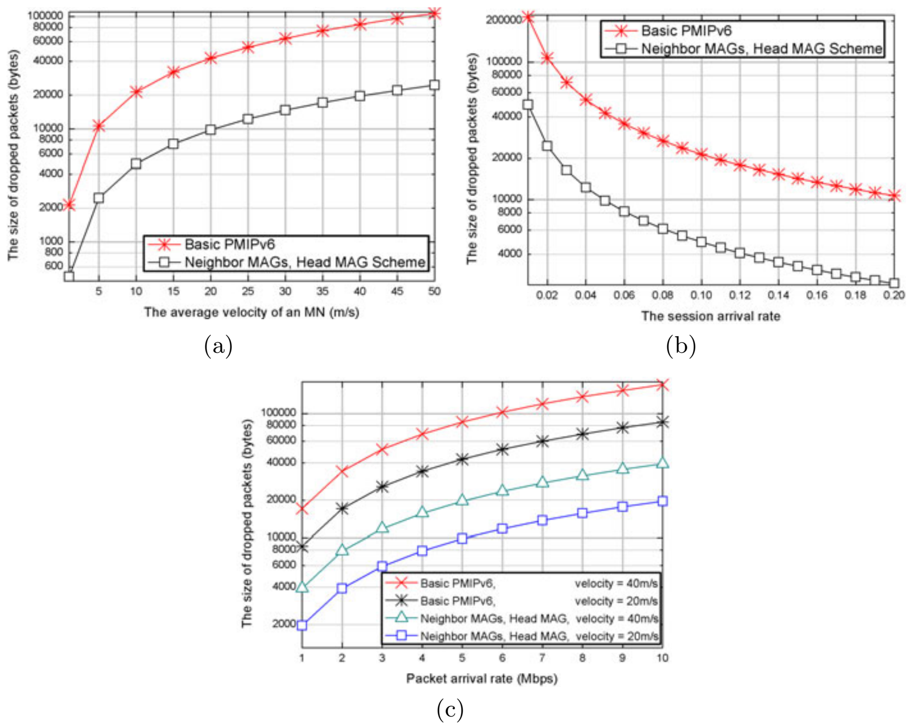


Fig. 5. Packet Loss: (a) the impact of the average velocity of an MN, (b) the impact of the session arrival rate, (c) the impact of the packet arrival rate

and 40 m/s, and λ_s is 0.05. All schemes increase the packet loss when the packet arrival rate is increased. However, the proposed scheme has lower packets loss than the basic PMIPv6 when the velocity of MN is faster than basic PMIPv6.

6 Conclusion

Basic PMIPv6 does not consider packet loss during handover. Many schemes have been proposed to solve the problem, but none of the existing schemes can solve the problem effectively. This paper proposed the seamless handover scheme using the Head MAG. Our proposed scheme can prevent packet loss during the MN's handover. The scheme does not use a L2 message, so the scheme does not violate the principle of PMIPv6. In addition, our proposed scheme provide more reliable handover service to an MN using the Head MAG.

In this paper, we evaluate performance using analytical model. We compare the signaling cost and packet loss with the basic PMIPv6 and Neighbor MAGs scheme during MN's handover. Our proposed scheme increased the signaling cost, but the packet loss is decreased effectively. Our proposed scheme reduces the signaling cost by over 30% compared to Neighbor MAGs scheme and the packet loss by over 78% compared to the basic PMIPv6 scheme. In addition, our proposed scheme decreased network topology overhead and provide seamless and reliable service during MN's handover.

Acknowledgements. This research was supported in part by MKE and MEST, Korean government, under ITRC NIPA-2011-(C1090-1121-0008), WCU NRF (No. R31-2010-000-10062-0), and Future-based Technology Development Program through the NRF funded by the MEST (No. 2010-0020727), respectively.

References

1. Johnson, D., Perkins, C., Arkko, J.: Mobility Support in IPv6. IETF RFC 3775 (2004)
2. Gundavelli, S., Leung, K., Devarapalli, V., Chowdhury, K., Patil, B.: Proxy Mobile IPv6. IETF RFC 5213 (2008)
3. Yokota, H., Chowdhury, K., Koodli, R., Patil, B., Xia, F.: Fast Handovers for Proxy Mobile IPv6. IETF RFC 5949 (2010)
4. Obele, B., Kang, M.: Mobility Management: A Proactive QoS-Aware Proxy MIP with improved Handover Latency for End-to-End QoS provisioning in a Proxy MIP domain. In: IEEE International Conference on Advanced Communication Technology, pp. 1867–1869 (2009)
5. Hwang, H., Kim, J., Lee, J., Lee, K.: Fast Handoff Scheme Using Multicast Group for Intra-domain in PMIPv6 Networks. In: IEEE Consumer Communications and Networking Conference, pp. 9–12 (2010)
6. Kang, J., Kum, D., Li, Y., Cho, Y.: Seamless Handover Scheme for Proxy Mobile IPv6. In: IEEE Wireless and Mobile Computing, pp. 410–414 (2008)
7. Zhang, X., Castellanos, J., Campbell, A.: P-MIP: Paging Extensions for Mobile IP. In: ACM Mobile Networks and Applications, pp. 127–141 (2002)
8. Malaya, C., Pierre, S.: An Analytical Framework for Performance Evaluation of IPv6-Based Mobility Management Protocols. IEEE Transactions on Wireless Communication 7(3), 972–983 (2008)
9. Lee, J., Ernst, T., Chung, T.: Cost analysis of IP mobility management protocols for consumer mobile devices. IEEE Transactions on Consumer Electronics 56(2), 1010–1017 (2010)
10. Kong, K., Lee, W., Han, Y., Shin, M., You, H.: Mobility Management for ALL-IP Mobile Network: Mobile IPv6 VS. Proxy Mobile IPv6. IEEE Wireless Communications 15(2), 36–45 (2008)

A Hole Detour Scheme Using Virtual Position Based on Residual Energy for Wireless Sensor Networks*

Zeehan Son[1], Myungsu Cha[1], Min Han Shon[1], Moonseong Kim[2], Mihui Kim[3], and Hyunseung Choo[1]

[1] School of Information and Communication Engineering
Sungkyunkwan University, Suwon, 440-746, Korea
{szhno85,chams,minari95,choo}@skku.edu
[2] Information and Communications Examination Bureau,
Korean Intellectual Property Office, Daejeon, 302-701, Korea
moonseong@kipo.go.kr
[3] Department of Computer Engineering,
Hankyong National University, Anseong, 456-749, Korea
mhkim@hknu.ac.kr

Abstract. Wireless Sensor Networks (WSNs) consist of a large number of low powered nodes that need to operate for months unattended. Since modern WSNs are used in various applications, their topology is becoming complicated. Due to limited precision of deployment, holes may occur in the network, which often lead traditional Greedy Forwarding algorithms to fail. Thus, bypassing the holes is one of the important issues for WSNs. Since each node has limited energy, its energy consumption needs to be optimized to prolong network lifetime. In the well-known Virtual Position (ViP) scheme, each node routes data using virtual positions instead of actual geographic positions to improve the packet delivery rate. A Hole-bypassing Routing with Context-awareness scheme achieves balanced energy consumption by changing current path to one of the candidate paths, based on the residual energy of nodes. However, this scheme tends to extend the size of holes. Since existing hole detour schemes that do not consider efficient energy consumption, they cause imbalanced energy consumption and make network lifetime relatively shorter than other hole detour schemes. Similar to ViP, our scheme uses virtual positions to bypass holes. However, the virtual positions are computed using both geographic positions and the residual energies of neighbor nodes. Our approach outperforms the ViP scheme in terms of network lifetime and hole extension.

Keywords: Hole detouring, greedy forwarding, geographic routing, virtual position.

* Corresponding author.

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 193–204, 2011.

1 Introduction

The early routing protocols in wireless sensor networks (WSNs) are on-demand routing, which uses a flooding procedure to find a route from a source to a destination. In these flooding schemes, data packets are flooded across the whole network to find its destination. This type of flooding is known to have a large overhead. To overcome the overhead problem, geographic routing was proposed. In geographic routing, every node uses GPS or another localization method to determine the positions of destination and neighbor nodes. In geographic routing, nodes just use their neighbor positions instead of the whole network topology information to transmit data packets, thereby obtaining good energy efficiency and extendability. In geographic routing, a source node piggybacks the destination node's position information in the packet and then uses original greedy forwarding [1], which sends the data packet to a node closest to the destination. This process is repeated until the packet arrives at the destination. But geographic routing suffers from local minimum problems.

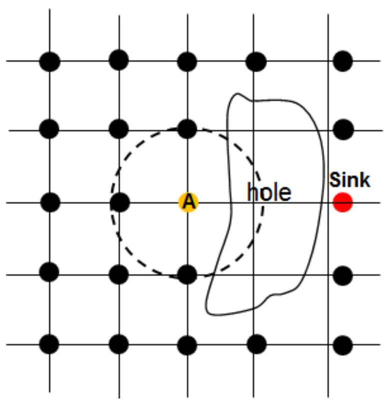


Fig. 1. Local minimum phenomenon

As WSNs are widely used in various applications, their deployments are becoming more complicated. In many applications, sensor nodes are simply scattered in an area and left unattended. Thus, either void in deployment or node failure can cause routing holes in the network, which often results in failures in geographic routing [2, 3]. The local minimum phenomenon is illustrated in Fig. 1. This phenomenon occurs when a packet is forwarded to a node like A in Fig. 1. Since there is no direct neighbor closer to the destination than Node A itself, node A receives all of the destination's traffic packets. This depletes node A's energy more quickly than other nodes. Uneven power consumption of sensor nodes can also lead to new routing holes in the later lifetime of WSNs [4].

A lot of hole-bypassing schemes were proposed to resolve the local minimum problem. One of the hole bypassing schemes is known as Greedy Perimeter Stateless Routing (GPSR) [5]. In this scheme, data transmitting mode is switched from greedy forwarding to perimeter mode whenever a data packet faces a local minimum during transmission. When the perimeter mode resolves the local minimum problem, the mode turns back to greedy forwarding. Jia *et. al* proposed a Hole Avoiding in advance Routing Protocol (HAIR) scheme. In this HAIR scheme, when a node faces local minimum it asks its neighbors to mark itself as a hole node [6]. Using this way, the data packets are sent to non-hole nodes. A Hole-bypassing Routing with Context-awareness (HobyCan) is proposed in [7]. In HobyCan, several pre-made paths around the hole are used depending on context information like residual energy of nodes. Greedy forwarding with Virtual position (Greedy-ViP) uses virtual positions of nodes to transmit data packets in greedy forwarding instead of nodes' geographic positions [4]. GPSR and Greedy-ViP are known as low-cost and efficient detouring methods, but these schemes do not consider the energy consumption of each node. Consequently, holes in the network spread over time, which is a serious problem. In GPSR and Greedy-ViP, a chosen optimal path is used to transmit the data packet every time. Therefore, intermediate nodes in the optimal path deplete their energy for transmission, which results in extension of the hole. In the worst case, the entire network may be broken.

In this paper, we propose a scheme that computes the virtual positions of nodes based on their residual energy to simultaneously resolve hole problems and extend network lifetime. Our scheme, Greedy Forwarding with Residual Energy based Virtual Position (EViP), is a hole-bypassing scheme which considers the residual energy of each node. In this scheme, a node calculates its virtual position considering the residual energies of the neighbor nodes. Therefore, virtual positions are directed towards nodes having more residual energy. Our contribution is as follows: EViP makes equitable energy consumption among nodes and improves network lifetime by 15%.

The remainder of the paper is organized as follows. Section 2 presents GPSR, HobyCan and ViP, while Section 3 introduces the proposed scheme. Section 4 compares the results between our scheme and other schemes. Concluding remarks and further suggestions are found in Section 5.

2 Related Work

B. Karp and H. T. Kung proposed Greedy Perimeter Stateless Routing (GPSR) [5], which is one of the fundamental protocols based on planar graphs. GPSR uses greedy forwarding and switches to perimeter routing mode when reaching a local minimum. Data packets are forwarded along the edge counterclockwise on the face of a planar graph. After bypassing the local minimum, it turns back to Greedy forwarding. GPSR uses only one optimal path each time to send packets. This causes uneven energy consumption among intermediate nodes in the optimal path, which leads to the extension of holes in the network.

The HobyCan protocol constructs multiple detour paths for a hole [7]. For packet routing around the hole, a suitable path can be dynamically determined from the set of detour paths. As a result, energy consumption is fairly distributed

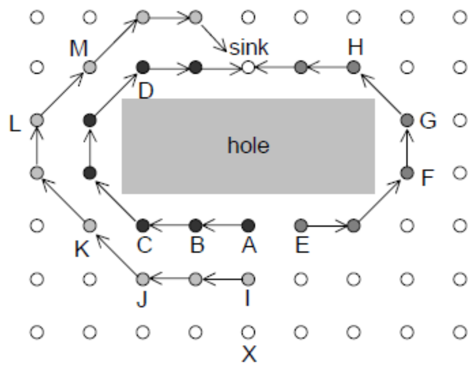


Fig. 2. HobyCan protocol

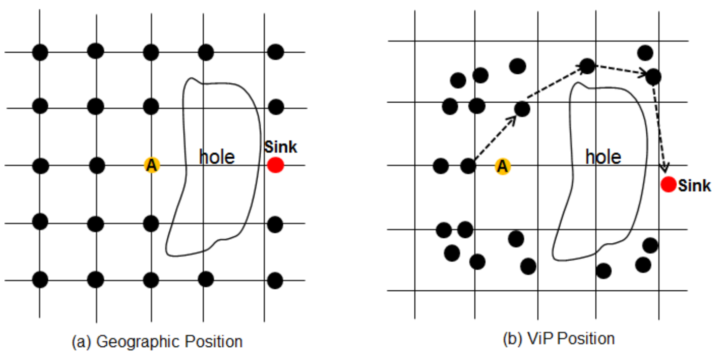


Fig. 3. ViP Protocol

among more nodes on extra detour paths. Nodes on detour paths are called detour nodes.

As shown in Fig. 2, when a local minimum occurs (at node A), a new detour path is constructed. New detour paths for the same hole are indexed with incrementing numbers. When the predefined number of detour paths is 3, disjoint detour paths $A - B - C - D - \bullet\bullet\bullet - Sink$, $E - F - G - H - \bullet\bullet\bullet - Sink$, and $I - J - K - L - M - \bullet\bullet\bullet - Sink$ are obtained using the same method [7].

J. You *et al.* solves the hole problem with ViP. ViP uses virtual positions of the nodes instead geographic positions [4]. This improves the success rate of greedy forwarding in sparse networks or in networks with some obstacles. When the network topology is created, all nodes calculate their virtual positions using Eq. (1). In other words, the virtual position of a node is the average of one-hop neighbors' coordinates. A data packet is forwarded using virtual positions instead of geographic positions.

$$(x_A^{'}, y_A^{'}) = ((x_{A,1} + x_{A,2} + ... + x_{A,n})/n, (y_{A,1} + y_{A,2} + ... + y_{A,n})/n) \qquad (1)$$

Fig. 3 illustrates the Greedy-ViP. The geographic position of node A is at the local minimum, so after computing the virtual position, the distance value of node A becomes greater than its previous one, causing its neighbors to ignore it when routing packets. All nodes calculate their virtual positions before sending data packets. This scheme improves the success rate of greedy forwarding, but it does not consider equitable energy consumption among nodes.

3 Detour Scheme Considering Residual Energy

This section explains how to calculate virtual positions in EViP. An overview of the EViP is presented in Subsection 3.1. Setting up method of virtual positions is illustrated in Subsection 3.2. Condition updating and data forwarding are explained in Subsection 3.3.

3.1 Greedy Forwarding with Residual Energy Based Virtual Position (EViP)

The proposed scheme tries to settle the hole problem using virtual positions and spreads the energy consumption among several paths to increase energy efficiency. In ViP, all nodes in the network have to calculate their virtual positions before routing data. This can generate an overhead problem and consume lots of energy. EViP has one main difference from other existing hole-bypassing schemes. In EViP, when a node calculates its virtual position, the residual energies of neighbors are considered. We give more weight to nodes having more residual energy than others during the calculation of virtual positions. The new virtual position will be directed towards nodes with more residual energy. When we use this method, the current node, which has a data packet, will choose the next hop with more residual energy than other nodes. Therefore, a new path will be made with new nodes having more energy and the data packet is efficiently transmitted to the destination. As a result, EViP help prolong network lifetime.

3.2 Virtual Position Setup

In EViP, all nodes calculate their virtual positions before forwarding data packets. When the network topology is made, nodes calculate the average of their one-hop neighbor coordinates and forward a packet to a destination. At first, the residual energy of every node is one, thus the virtual position of every node is the same as in ViP. After sending several packets, if the residual energy of a node is lower than a threshold, it will calculate a new virtual position using Eq. (2).

$$(x'_A, y'_A) = ((\frac{\alpha_{A,1}x_{A,1} + ... + \alpha_{A,n}x_{A,n}}{\alpha_{A,1} + ... + \alpha_{A,n}}), (\frac{\alpha_{A,1}y_{A,1} + ... + \alpha_{A,n}y_{A,n}}{\alpha_{A,1} + ... + \alpha_{A,n}})) \quad (2)$$

Eq. (2) is a weighted average expression appearing in many references [8, 9]. Each node has 10,000 units of energy at the beginning. However, in this paper

α is the residual energy weight ($0 \leq \alpha_{A,i} \leq 1$); $x_{A,1}$ is the x coordinate of the first neighbor of node A; and $y_{A,1}$ is the y coordinate of the same node.

EViP calculates the average of all coordinates weighted with residual energy. After consuming energy to forward data packets, the remaining energy of a node will reach a threshold. At this moment, it uses Eq. (2) to calculate its virtual position. The residual energy of node A can be lower than other neighbors, so its new virtual position will be directed toward nodes having more residual energy. Because the energy weight of node A is smaller than its neighbors, it can make a new path with new nodes having more residual energy.

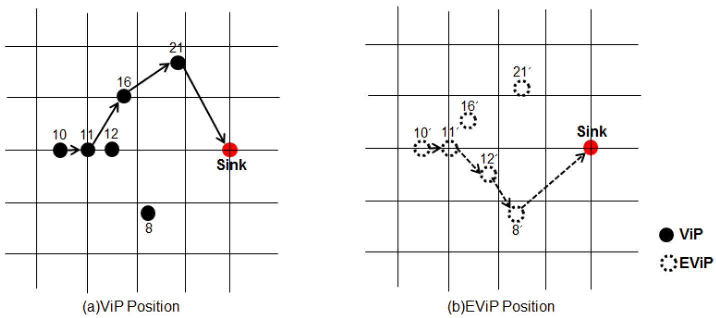


Fig. 4. Node deployment using the weight in EViP

Fig. 4 shows an example of the virtual position updating procedure. When all nodes have full energy in EViP, the virtual position of sensor nodes are deployed like ViP in Fig. 4 (a). When a data packet is transmitted along the path $10 - 11 - 16 - 21 - sink$, node 16 will consume its energy to send the data packet to node 21. Thus, node 16 will have less residual energy than node 8. Therefore, when EViP calculates virtual positions, the weight of Node 16 is less than that of node 8. Finally, the virtual position of node 12 is shifted to node $12'$. In greedy forwarding, the current Node 11 will choose node $12'$ because node $12'$ is closer to the destination than node 16. Thus, the data packet will be sent to node $12'$ and transmitted to the destination using the dotted line $10'$-$11'$-$12'$-$8'$-sink, as shown in Fig. 4 (b).

3.3 Virtual Position Updating and Data Forwarding

In EViP, the residual energy of each node will be changed by the data forwarding process. Therefore, the new path needs to be constructed considering residual energy of each node. The new path will be constructed by updating a virtual position near the nodes that has more residual energy. The nodes update their virtual positions whenever their residual energy is under a threshold.

If the residual energy of node A is lower than the threshold, it broadcasts a message to its one-hop neighbor nodes to let them know that its residual energy is under the threshold. The nodes that received the message change the virtual position and broadcast the new virtual position to their one-hop neighbors in

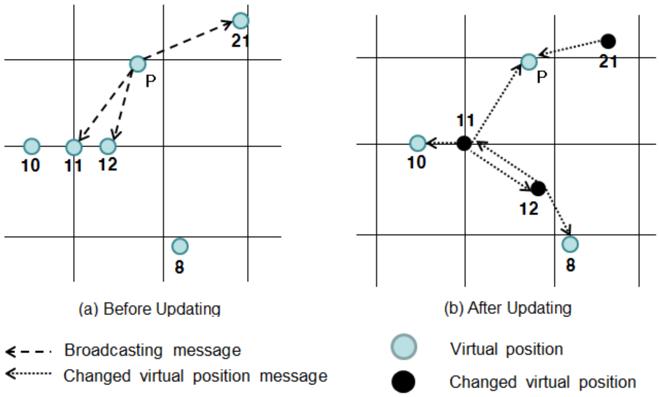


Fig. 5. Virtual position updating condition

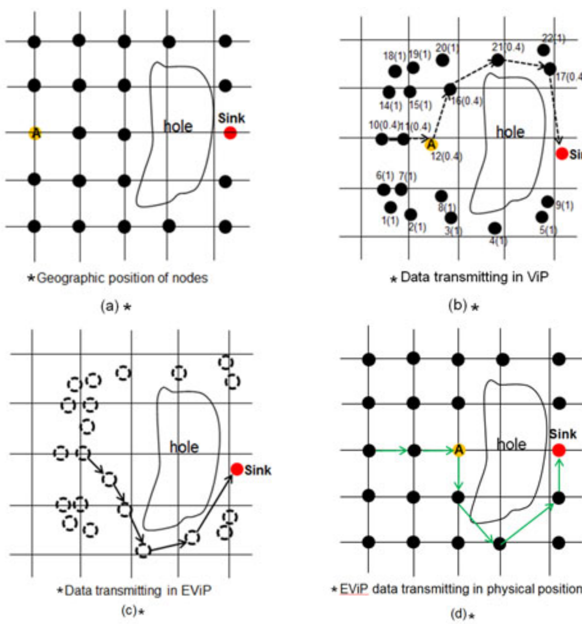


Fig. 6. Data forwarding in EViP

turn. With this method, one node only updates its virtual position locally and does not have to calculate the virtual positions of all nodes in the network like in ViP. Therefore it has smaller overhead than ViP.

For example, Fig. 5 illustrates the virtual position updating condition. If the residual energy of node P is less than the energy threshold, it broadcasts a message to its one-hop neighbor nodes. The nodes that receive the message change the virtual positions and broadcast the new virtual positions to one-hop neighbors.

In this scheme, once the EViP positions are obtained, the data packets are transmitted via greedy forwarding. When the nodes are in real geographic position, data packets will be stuck as they meet the local minimum or some obstacle in the network. But EViP has hole information during calculation of the virtual position, so it could easily bypass the hole. Fig. 6 shows how the data packet is transmitted to the destination. Fig. 6 (a) shows geographic positions of all nodes. The local minimum problem may occur. Node A attempts to send a packet to the sink, and all nodes in the network calculate their virtual positions. When all nodes have full energy in EViP, the virtual position of sensor nodes are deployed like ViP, and packets are sent using greedy forwarding (dotted line) as in Fig. 6 (b). The number in the brackets shows the nodes' residual energy. The nodes' residual energy is 1 at the beginning. After sending packets, if the nodes see their residual energy drop lower than a threshold, they will change their virtual position. Every node calculates their EViP positions as shown in Fig. 6 (c). After calculating EViP, the data packet will be transmitted like a solid line in Fig. 6 (c). The real data forwarding works as in Fig. 6 (d).

4 Performance Evaluation

We evaluate our scheme in a 1,000 x 1,000 square meter area and generated a hole randomly [7]. We randomly select source nodes at the southwest region, and sink nodes at the northeast region to create a hole effect during data transmission [10]. The initial energy level of each node is set to 10,000. The simulation was based on the energy consumption model observed in [8], that is, the ratio of energy consumed in receiving and transmitting is 19.7 and 17.4 [11], respectively. Each node in the figure is the result of 1,000 simulation runs.

4.1 Speed of Hole Spread as Hole Size

In this simulation, we change the hole size from 10% to 50% of the network area to compare the speed of hole spread. The transmission range is 100m [12]. We

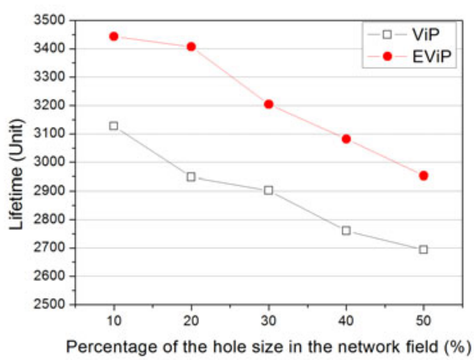


Fig. 7. Network lifetime based on creation probability of hole

have measured the network life time when the network is disconnected [13,14]. Fig. 7 shows that the proposed EViP scheme has a longer lifetime than ViP. The reason is that ViP uses only the optimal path when transmitting a data packet. Therefore, the network hole gets bigger. But EViP makes a new path with nodes that have more residual energy. Thus, the hole spread speed would be slower and the network lifetime is increased.

4.2 Speed of Hole Spread as Number of Nodes

Fig. 8 shows the network lifetime based on different hole size and number of nodes. The simulation environment is similar to that of the previous study. We only change the number of nodes. As shown in Fig. 8, our proposed EViP scheme has much longer lifetime than ViP in simulations with hole creation probabilities of 10%, 20%, and even 30%. The reason is almost the same as previous simulations in that EViP makes new paths besides the optimal path. When time passes, EViP will make more new paths with new nodes that have high residual energy. But EViP shows the same network lifetime when the number of nodes is 1,000. This increases lifetime because if the ratio of holes increases, the node density in the network field except holes is denser, so EViP makes new paths with many nodes and slows down the speed of hole spread.

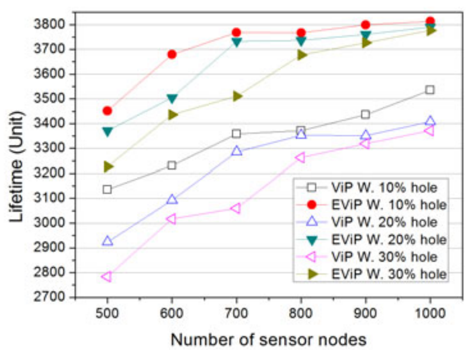


Fig. 8. Speed of hole spread with different number of nodes

4.3 Lifetime of WSNs

Fig. 9 shows the network lifetime of ViP and EViP schemes. The ViP scheme uses one path during transmission from source node to sink node. So nodes located in the path deplete their energy fast. Eventually, nodes located in the path exhaust energy quickly. Initially, the EViP and ViP schemes have the same virtual coordinates because every node has the same energy in the network. However, if any node reaches its threshold, its neighbor nodes, which also reach thresholds, recalculate their virtual positions considering the energy of neighbor nodes. Recalculation of virtual positions excludes nodes that have lower energy

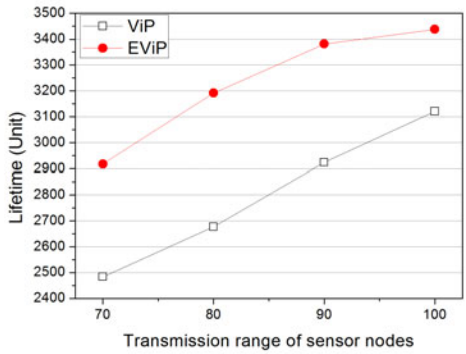


Fig. 9. Lifetime of WSNs

from routing paths. Thus a node which has lower energy than its neighbors saves that energy. The EViP scheme achieves 14% longer network lifetime than the ViP scheme. Fig. 9 also shows that if each node has a longer transmission range, the lifetime of the network is extended. If a sensor node has a shorter transmission range, the probability of the local minimum phenomenon is increased during routing. Therefore, it selects some specific sensor nodes during routing, and those nodes use up their energy quickly.

4.4 End-to-End Deliver

Fig. 10 shows the ratio of the average transmission time in the ViP and EViP schemes. We repeat our experiments with different transmission ranges. However, for every case of transmission, the ratio of the average routing time between ViP and EViP schemes is similar. This ratio similarity is because only neighbor nodes

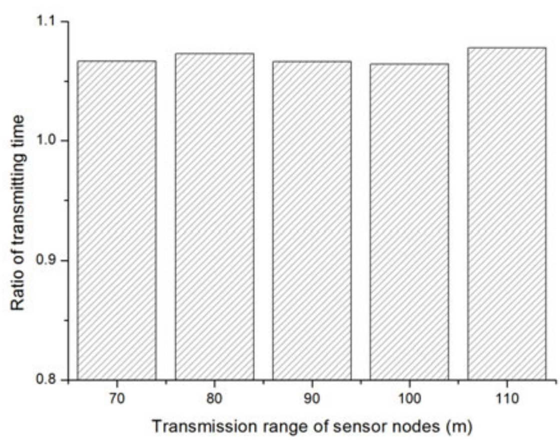


Fig. 10. Ratio of transmitting time

of a node which reaches a threshold recalculate their virtual positions. Therefore, most sensor nodes do not change their virtual positions. Also, the EViP scheme has a longer average routing time, about 1.07 times longer than the ViP scheme.

However, energy efficiency of nodes is a critical problem in WSNs, because wireless sensor networks consist of a large number of low powered nodes that need to operate for months unattended. In order to sustain sensors to run for a long period of time with limited energy, it is necessary to save energy in sensor operations. Thus a slightly longer transmission delay is acceptable in exchange for node energy efficiency.

5 Conclusion

Energy efficiency of nodes and routing around holes in the network are very important problems in wireless sensor networks. In this paper, we propose the EViP scheme to simultaneously solve both problems. The EViP scheme uses a weighted average method considering residual energy of neighbor nodes when each node calculates its virtual position. We simultaneously solved inequitable energy consumption in existing hole bypassing schemes and hole problems in energy efficient routing schemes. Therefore, the virtual position of nodes is determined based on how much energy a node has compared to its neighbors. Such considerations efficiently bypass holes and prevent holes from growing. Simulation results show that our proposed scheme has 15% longer network lifetime than the ViP scheme. And since our scheme is distributed, it has the strong point of scalability. We will compare EViP and routing schemes which consider energy balance applying ViP in further research.

Acknowledgment

This research was supported in part by MKE and MEST, Korean government, under ITRC NIPA-2011-(C1090-1121-0008), WCU NRF (No. R31-2010-000-10062-0) and PRCP(2010-0020210) through NRF, respectively.

References

1. Finn, G.G.: Routing and addressing problems in large metropolitan-scale internetworks. Technical Report ISI/RR-87-180, Information Sciences Institute (1987)
2. Warneke, B., Last, M., Liebowitz, B., Kristofer, Pister, S.J.: Smart Dust: Communicating with a Cubic-Millimeter Computer. Computer Magazine 34(1), 44–51 (2001)
3. Ahmed, N., Kanhere, S.S., Jha, S.: The hole problem in wireless sensor networks: a survey. ACM SIGMOBILE Mobile Computing and Communications Review 9(2) (2005)
4. You, J., Lieckfeldt, D., Han, Q., Salzmann, J., Timmermann, D.: Look-ahead Geographic Routing for Sensor Networks. In: Proceedings of PERCOM (2009)

5. Karp, B., Kung, H.T.: GPSR: Greedy perimeter stateless routing for wireless networks. In: Proceedings of MobiCom, pp. 243–254 (2000)
6. Jia, W., Wang, T., Wang, G., Guo, M.: Hole Avoiding in Advance Routing in Wireless Sensor Networks. In: Proceedings of IEEE WCNC (2007)
7. You, J., Lieckfeldt, D., Reichenbach, F., Timmermann, D.: Context-aware Geographic Routing for Sensor Networks with Routing Holes. In: Proceedings of IEEE WCNC (2009)
8. Sheng, X., Hu, Y.: Maximum Likelihood Multiple-Source Localization Using Acoustic Energy Measurements with Wireless Sensor Networks. IEEE Transactions on Signal Processing 53(1) (2005)
9. Niu, R., Varshney, K.: Target Location Estimation in Sensor Networks with Quantized Data. IEEE Transactions on Signal Processing 54(12) (2006)
10. Zhu, J., Chen, S., Bensaou, B., Hung, K.-L.: Tradeoff between lifetime and rate allocation in wireless sensor networks: a cross layer approach. In: Proceedings of IEEE INFOCOM (2007)
11. Texas Intruments. 2.4 GHz IEEE 802.15.4 / ZigBee-Ready RF Transceiver, Rev. B (2007), http://focus.ti.com/docs/prod/folders/print/cc2420.html
12. Tang, J., Xue, G.: Node-disjoint path routing in wireless networks: tradeoff between path lifetime and total energy. In: Proceedings of IEEE ICC, pp. 3812–3816 (2004)
13. Maleki, M., Dantu, K., Pedram, M.: Lifetime prediction routing in mobile ad hoc networks. In: Proceedings of IEEE WCNC (2003)
14. Xu, Y., Heidemann, J.S., Estrin, D.: Geography-informed energy conservation for ad hoc routing. In: Proceedings of MOBICOM, pp. 70–84 (2001)

An Improved MPLS-MOB for Reducing Packet Loss in NGN

Myoung Ju Yu[1], Seong Gon Choi[1,*], Sung Won Sohn[2]

[1] College of Electrical & Computer Engineering, Chungbuk National University (CBNU), 410 Seongbong-ro, Heungdeok-gu, Cheongju, Chungbuk 361-763, Republic of Korea
[2] Electronics and Telecommunications Research Institute (ETRI), 138 Gajeongno, Yuseong-gu, Daejeon, Republic of Korea
{mjyu,sgchoi}@cbnu.ac.kr, swsohn@etri.re.kr

Abstract. We propose an improved MPLS-MOB scheme for seamless service in Next Generation Network (NGN). The existing MPLS-MOB scheme delivers signaling messages by using L2.5 switching via MPLS LSP for low handover latency. However, the MPLS-MOB scheme does not consider a solution to prevent packet loss during handover. The proposed scheme reduces packet loss by adding some process before and after connecting to new location on the existing MPLS-MOB. So, it can overcome problems from existing schemes, such as MIPv6, FMIPv6 and the existing MPLS-MOB. We analyze each scheme numerically and verify that the proposed scheme has better performance than others on handover latency and packet loss.

Keywords: NGN, Fast Handover, MPLS, Handover Latency, Packet Loss.

1 Introduction

The demand for next generation wireless systems, which intend to support seamless communication and various multimedia services, is increasingly accelerated along with the rapid development of mobile terminal and wireless technology. With an explosive growth of users using wireless communication networks, the issue of IP mobility management technology is on the rise. Therefore, a variety of studies for seamless mobility including service continuity have progressed [1]-[4].

For these reasons, the IETF has proposed various mobility-enabling solutions, such as Mobile IP (MIP) and Fast Mobile IP (FMIP) for supporting mobility in almost all packet-based wireless mobile systems [2]-[4]. MIP (MIPv4 and MIPv6) is a well-known IP mobility support protocol. However, it has some serious problems such as considerable handover latency, high packet loss, power consumption, signaling overhead. In case of handover, a Mobile Node (MN) has to perform a sequence of procedures such as, link layer scanning, stateless address auto-configuration, duplicate address detection and Binding Updates (BU). Until completing the

* Corresponding author.

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 205–216, 2011.

above-mentioned processes, the MN cannot receive and transmit any packet. The time interval is handover latency and packet loss occurs during the period [5].

For solving the problems such as handover latency and packet loss, many studies including FMIP [4] and MPLS-MOB (MPLS-based Mobility Management Scheme) [1] have been progressed. FMIP allows an AR to provide services to an MN in order to anticipate the layer 3 handover. For this, the FMIP uses FMIP tunnel which helps the PAR forward packets during layer 2 switching and until MIP registration to complete. However, the FMIP have still some problems such handover delay and packet loss as it takes much time to carry out tunnel management procedures according to the MN's operating system and to update IPv6 routing information. The packet loss can be enlarged due to operating time delay [5]. The MPLS-MOB delivers signaling messages with L2.5 switching via MPLS LSP and initiates MM during L2 association for low handover latency. However, the MPLS-MOB does not consider a solution to prevent packet loss during handover. So, it needs more study to reduce packet loss.

In this paper, we propose an improved MPLS-MOB scheme that supports small packet loss as well as low latency time during handover. It is based on [1], precedent study. The proposed scheme improves the existing MPLS-MOB by adding some process to avoid packet loss. When a MN's movement is perceived, old network (current network) requests to buffer packets to the MN to Corresponding Node (CN). The packets sent to the MN from the CN before the CN receives the request are buffered on the old network. When the MN requests its location update to MICS, MICS sends response message to old network as well as new network. The response message plays a role as trigger which indicates to deliver the buffered packets from old network to new network. For performance analysis, we evaluate handover latency and packet loss based on analytical model [6]-[8]. For comparison, MIPv6 [2], FMIPv6 [4] and MPLS-MOB [1] are analyzed as well. The numerical results show that the proposed scheme outperforms others.

The remainder of this paper is organized as follows. Section 2 introduces MIPv6, FMIPv6 and MPLS-MOB as the related works. In Section 3, we describe the detailed procedures of the proposed scheme. Section 4 presents the analytical models and numerical results. Finally, Section 5 concludes this paper.

2 Related Works

This section introduces MIPv6, FMIPv6 and MPLS-MOB as the related works.

2.1 Mobile IPv6

The MIPv6 specification enables hosts to change their point of attachment to the Internet whilst not breaking existing application sessions.

While a MN is attached to its home network, it is able to receive packets destined to its Home Address (HoA), and being forwarded by means of conventional IP routing mechanisms. When the MN moves into a new network, its movement is detected and a new association is made with foreign agents in the new domain. To obtain its CoA, a MN uses the IPv6 protocols for address auto-configuration and neighbor discovery. Once configured with a CoA, the MN needs to send a BU message to its Home Agent (HA) to register its current CoA. The first time a CN needs to communicate with the MN, it sends the packets to the MN's HoA. The HA is then encapsulating the packets and forwards them to the MN's CoA, where they are de-capsulated by the corresponding mobility agent and forwarded to the MN [9].

Upon a new association, the MN transmits BUs to its HA and the communicating CNs for them to associate the MN's HoA with the new CoA. When the MN received a Binding Acknowledgement (BA) from the HA, it is once again routable at the new location. After establishing a binding update with the HA, the MN initiates a return routability procedure with the CN. The purpose of this procedure is to enable the CN to make sure that the MN is in fact addressable at its claimed CoA as well as at its HoA. Only with this assurance is the CN able to accept BUs from the MN which would then instruct the CN to direct that MN's data traffic to its claimed CoA. From then on, CNs will use IPv6 Routing headers for forwarding the packets to the MN. These packets have as destination address the MN's CoA. The 'HoA' information is also included in the routing header to preserve transparency to upper layers and ensure session continuity. In the reverse direction, datagrams sent by the MN can be delivered to their destination using standard IP routing, without having to go through the HA. Packets have as a source address the host's CoA while the HoA is also included for the same reasons as above. By following this process, MIPv6 has inherent route optimization and does not suffer from Triangular routing problems as its predecessor [9].

When, however, the MN reaches its current wireless subnet, it has to perform a sequence of procedures such as movement detection and new CoA configuration. Also, when a MN changes its point of attachment to the network, the MN usually has disconnected from the current network before connecting to the new network and thus there is a time interval in which the MN has lost connectivity to the Internet. That is, it takes a long time to process handover and there is much packet loss during handover, since there are many signaling messages via wireless link [10].

2.2 Fast MIPv6

The FMIPv6 is designed to enable an MN to rapidly detect its movements and to obtain a prospective IP address with a new AR while being connected to a current AR. This protocol also offers the MN an opportunity to utilize available link layer triggers to accelerate network layer handover. Hence, delays due to network prefix discovery and new CoA configuration are completely eliminated during handoff. Moreover, a bidirectional tunnel is setup between the Previous AR (PAR) and the New AR (NAR) to avoid packet drops. The PAR binds an MN's previous CoA with its new CoA. Therefore, packets addressed to the MN are intercepted by the PAR, tunneled to the NAR [11].

While an MN is connected to its PAR and is about to move to a NAR, FMIP requires that the MN to obtain a new CoA at the NAR while still being connected to the PAR, the MN to send a BU message to its PAR to update its binding cache with the MN's NCoA and the PAR to start forwarding packets destined for the MN to the NAR. Either the MN or the PAR may initiate the Fast Handover procedure by using the L2 trigger. If the L2 trigger is received at the MN, the MN will initiate L3 handover by sending a Router Solicitation for Proxy (RtSoPr) message to the PAR. On the other hand, if the L2 trigger is received at the PAR, then the PAR will transmit a Proxy Router Advertisement (PrRtAdv) message to the appropriate MN. The MN obtains a new CoA, while still connected to the PAR, by means of router advertisements from the NAR. The PAR validates the MN's new CoA and initiates the process of establishing a bidirectional tunnel between the PAR and the NAR by sending a Handover Initiate (HI) message to the NAR. Then, the NAR verifies that its new CoA can be used on the NAR's link. Also, in response to the Handover Acknowledge message (HACK), the NAR sets up a host route for the MN's previous CoA (PCoA) and responds with a HACK message. When the MN receives a PrRtAdv message, it should send a Fast Binding Update (FBU) message. When the PAR receives an FBU message, it must verify that the requested handover is accepted by the NAR. Then it begins forwarding packets intended for PCoA to the NAR and sends a Fast Binding Acknowledgement (F-BACK) message to the MN. After the change of the link connectivity with the NAR, the MN sends an Unsolicited Neighbor Advertisement (UNA) message to the NAR. After that, the NAR starts to deliver buffered packets tunneled from PAR and buffered packets from CN directly [12].

Generally, the MN creates new tunnel, deletes old tunnel and updates routing information after receiving FBU and BU messages. However, while completing these tunnel management procedures, the MN cannot handle packets incoming and outgoing via new FMIP or MIP tunnel. In case of download, the packet loss could occur while the MN is in progress of establishing a new tunnel after receiving BA messages. In case of upload, packets could be lost as soon as receiving FBA while the MN establishes the FMIP tunnel. Also, upload packets could be lost while the HA establishes the new MIP tunnel after receiving the BU. The packet loss can be enlarged due to operating time delay [5].

2.3 MPLS-MOB

The MPLS-MOB supports low handover delay by delivering signaling messages via MPLS LSP and initiating MM during L2 association [1]. However, MPLS-MOB does not consider a solution to prevent packet loss during handover. So, it needs further study to reduce packet loss.

Figure 1 shows the network configuration of the proposed MM scheme. The Mobility Information Control Server (MICS), central address manager, manages MAC address, permanent IP address (IP_PA), and local IP address (IP_LA) of an MN as well as Handover Control Agent (HCA)'s IP address, and manages binding information related to communication between the MN and the Correspondent Nodes (CNs). The HCA, local address manager, manages MAC address, IP_PA, and IP_LA of an MN, and encapsulates packets for data transmission. The Access Point (AP) forwards an MN's MAC address to HCA when an MN enters into its area. The LSPs between HCAs and MICS are used to transmit only MM signaling message [1].

In case that an MN initially access to a network, when the MN enters into the AP#1 area, the AP#1 catches the MN's MAC address and then sends a Location Report message to the HCA#1. The HCA#1 creates a record for the MN in its Local Address Management Table (L-AMT), and sends a Location Registration message to the MICS, sending an Address Inform message to the MN. And the MICS creates a record for the MN in its Central Address Management Table (C-AMT). The MICS has MAC address and IP_LA of the MN, as well as the HCA#1's IP address. During the processing of the MICS, the MN sends an Address Inform ACK message to the HCA#1 in response to the Address Inform message from the HCA#1. When the HCA#1 receives the Address Inform ACK message, it sends an Address Update message to the MICS.

For data transmission, when the HCA#3 receives a packet toward the MN from the CN, it refers to the MN's IP_LA in its L-AMT. If it has no MN's IP_LA, it sends a Location Request message to the MICS. The MICS searches the MN's IP_LA, creates a record of IP_PA mapping about connection between the MN and the CN, sends a Location Response messages to the HCA#1 as well as the HCA#3. The HCA#3 encapsulates the packet with the destination address and the source address (CN's IP_LA), and the packet is tunneled from the HCA#3 to the MN.

In case of handover to another network, when the MN moves from the AN#1 to the AN#2, the AP#2 catches the MN's MAC address and sends a Location Report to the HCA#2. The HCA#2 creates a record for the MN in its L-AMT, writes the MN's MAC address and IP_LA, and sends a Location Registration message to the MICS, sending an Address Inform message to the MN. The MICS updates the record of the MN in its C-AMT, and sends a Location Response message to the HCA#2, while sending other Location Response message to HCA#3 that keeps the connection with the MN [1].

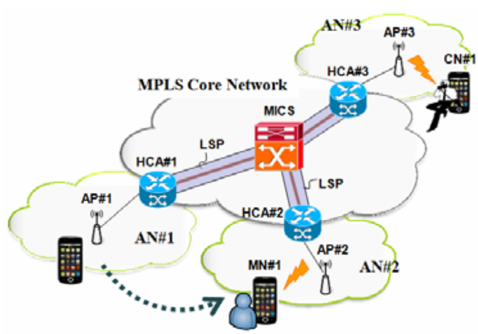


Fig. 1. Network configuration of MPLS-MOB

3 Proposed Scheme

This section presents the proposed scheme, improved MPLS-MOB. The proposed scheme is an extension of MPLS-MOB [1] to support small packet loss as well as low handover latency time.

As previously stated, the existing MPLS-MOB does not handle any solution to reduce packet loss. This issue causes service disruption during handover. To support seamless service, the proposed scheme suggests reducing packet loss which can be occurred from handover by adding some procedure on the existing MPLS-MOB.

When an MN is connected to its old HCA and is about to move to a new HCA, the old HCA requests to buffer packets addressed to the MN to a CN communicating with the MN. At that time, the packets sent to the MN before the CN receives the request arrives on the old HCA and then are buffered. The MN initiates to access to new HCA during L2 association and then obtains new IP address from the HCA. Each HCA prepares an IP address for an MN in advance. So, IP configuration process is not needed separately. When the MN requests its location update to MICS, the MICS updates its C-AMT and then respond to the new HCA and the CN's HCA. Also, the MICS sends a response message to the old HCA. The response message to the new HCA and the CN's HCA indicates tunnel establishment for packet delivery between two HCA. And the response message to the old HCA plays a role as trigger which starts forwarding the buffered packets destined for the MN to the new HCA. After that, the buffered packets on the old HCA are delivered to the MN via the new HCA. Also, the CN's HCA sends the buffered packets destined for the MN to the new HCA and then it delivers the packets to the MN.

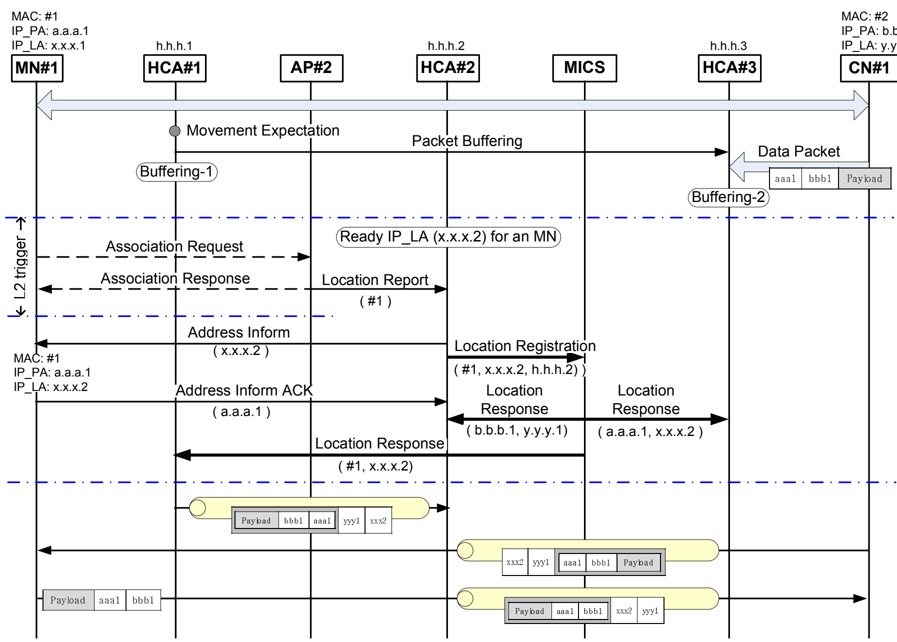


Fig. 2. Handover procedure of the proposed scheme

Figure 2 shows the message flow of handover in the proposed scheme. We assume that MN#1 and CN#1 are communicating after session establishment and the MN#1 tries to move from HCA#1 to HCA#2. When HCA#1 perceives the MN#1's movement, it requests to buffer packets for the MN#1 to HCA#3. After receiving the request, the HCA#3 starts buffering the packets. Also, when inflight packets which are sent to MN#1 from CN#1 are arrived on HCA#1, the HCA#1 buffers the packets.

When an MN moves to the AN#2, the AP#2 catches the MN's MAC address and sends a Location Report to the HCA#2. The HCA#2 creates a record for the MN in its L-AMT, writes the MN's MAC address and IP_LA, and sends a Location Registration message to the MICS, sending an Address Inform message (with MN's IP_LA) to the MN. The MICS updates the record of the MN in its C-AMT, and sends a Location Response message to the HCA#2, while sending other Location Response message to HCA#3 that keeps the connection with the MN. Also, it sends a Location Response message to the HCA#1 for triggering the delivery of the buffered packet to HCA#2. The MICS has MAC address and IP_LA of the MN, as well as the HCA#1's IP address. During the processing of the MICS, the MN sends an Address Inform ACK message (with MN's IP_PA) to the HCA#1 in response to the Address Inform message from the HCA#1. When the HCA#1 receives the Address Inform ACK message, it sends an Address Update message (with MN's IP_PA) to the MICS.

4 Performance Analysis and Results

In this section, we analyze the performance of the proposed scheme. Part of the analysis follows the derivation in [6]-[8]. For comparison, MIPv6 [2], FMIPv6 [4] and MPLS-MOB [1] is analyzed as well. We analyze the handover latency time and the number of lost packets during handover.

The handover latency is defined as the time interval from the time the MN loses the connection with the previous network until the time the MN receives or transmits packets by a new IP address on new network [6]. The handover latency consists of the movement detection delay, IP configuration delay, location update delay [7]. The parameter notations and value used for the analysis are defined in Table 1 as follows:

Table 1. Parameter values for performance analysis

	Parameter	Value
$T_{MD,MIP}$	The movement detection delay in MIP	50ms
$T_{AC,MIP}$	The auto-configuration delay in MIP	1000ms
t_{mr}	The delay between MN and AP	10ms
t_{ra}	The delay between AP and AR/HCA	2ms
t_{pn}	The delay between PAR and NAR	5ms
t_{HH}	The delay between different HCAS	5ms
t_{ah}	The delay between AR and HA	40ms
t_{HM}	The delay between HCA and MICS on MPLS-based network	7ms
t_{ac}	The delay between AR/HCA and CN, not via HA	40ms
t_{hc}	The delay between HA and CN	20ms
λ_p	Packet arrival rate	10 packets/s

1) MIPv6: As shown in [7], the binding update delay in MIPv6 ($T_{BU,MIP} = 4(t_{mr} + t_{ra}) + 2(t_{ah} + t_{ac})$) includes the time of the binding update delay to the HA (i.e., $2(t_{mr} + t_{ra} + t_{ah})$) plus the binding update delay to the CN (i.e., $2(t_{mr} + t_{ra} + t_{ac})$). On the other hand, in order to perform binding update procedure with the CN, the delay for the return routability (i.e., $T_{RR,MIP} = 2(t_{mr} + t_{ra} + t_{ah} + t_{hc})$) is additionally required prior to the binding update to the CN. Therefore, the total handover latency in MIPv6 ($D_{HO,MIP}$) can be expressed as follows:

$$D_{HO,MIP} = t_{mr} + T_{MD,MIP} + T_{AC,MIP} + T_{RR,MIP} + T_{BU,MIP} \tag{1}$$

Packets are lost before an MN register its new CoA with the HA. The number of lost packets during handover in MIP is

$$L_{HO,MIP} = \lambda_p \times (D_{HO,MIP} - (t_{mr} + t_{ra} + t_{ac})) \tag{2}$$

2) FMIPv6: In predictive mode, FMIPv6 informs the MN of the NAR's network prefix via the PAR and validating the uniqueness of the prospective CoA on the NAR prior to the MN's movement. FMIPv6 intends to eliminate the factors of the delay introduced by the IP-level movement detection and the address configuration procedures in MIPv6. However, actually, handover latency has been started since the PAR, upon receiving FBU, stops forwarding packets for the MN [7]. The MN can get the packets again after the NAR receives UNA. Also, the PAR supports for buffering packets after exchanging HI and HACK. Therefore, the packet loss could occur while the MN is in progress of establishing a new tunnel [4]-[5]. Since FMIPv6 basically operates based on the movement prediction, perfect prediction may be difficult in real environment, and thus it may also operate in reactive mode. Therefore, according to [7], the total handover latencies of FMIPv6-predictive mode ($D_{HO,FMIP-pre}$) and FMIPv6-reactive mode ($D_{HO,FMIP-rea}$) can be expressed as follows

$$D_{HO,FMIP-pre} = 4t_{mr} + 3t_{ra} + 2t_{pn} \tag{3}$$

$$D_{HO,FMIP-rea} = 3t_{mr} + 2(t_{ra} + t_{pn}) \tag{4}$$

In predictive mode of FMIPv6, packets are lost in PAR during exchanging HI and HACK. We do not consider the case that the coordination of fast handover signaling is not correct or packets are failed to the buffered in NAR or PAR. Therefore, the number of lost packets is

$$L_{HO,FMIP-pre} = \lambda_p \times 2t_{pn} \tag{5}$$

In reactive mode of FMIPv6, packets are lost in PAR before setting up the forwarding link between PAR and NAR. The packet loss is derived as

$$L_{HO,FMIP-rea} = \lambda_p \times (D_{HO,FMIP-rea} - t_{pn}) \tag{6}$$

3) MPLS-MOB: As shown in [1], handover latency time in the MPLS-MOB only includes location update delay. The MPLS-MOB does not have the movement detection delay by initiating MM during L2 association. Also, it does not include IP configuration delay because an IP address for an MN is allocated from DHCP server in advance. Therefore, the handover delay of the MPLS-MOB is derived as

$$D_{HO,MPLS-MOB} = 2t_{mr} + t_{ra} + 3t_{HM} \tag{7}$$

Packets are lost before an MN updates its new location with the MICS. The number of lost packets during handover in the existing MPLS-MOB is

$$L_{HO,MPLS-MOB} = \lambda_p \times (D_{HO,MPLS-MOB} - (t_{mr} + t_{ra} + t_{HH})) \tag{8}$$

4) Proposed Scheme (Improved-MPLS-MOB): handover latency time in the proposed scheme only includes location update delay like MPLS-MOB. After location update, the buffered packets destined for an MN are delivered from old HCA to new HCA. Therefore, the handover delay of the proposed scheme is derived as

$$D_{HO,proposed-scheme} = 2t_{mr} + t_{ra} + 2t_{HM} + t_{HH} \tag{9}$$

And then the number of lost packets during handover in the proposed scheme is

$$L_{HO,proposed-scheme} = \lambda_p \times (D_{HO,proposed-scheme} - (2t_{mr} + t_{ra} + 2t_{HM} + t_{HH})) \tag{10}$$

Figure 3 and 4 illustrate the handover latency and packet loss of the existing mobility protocols such as MIPv6, FMIPv6, MPLS-MOB as well as the proposed scheme.

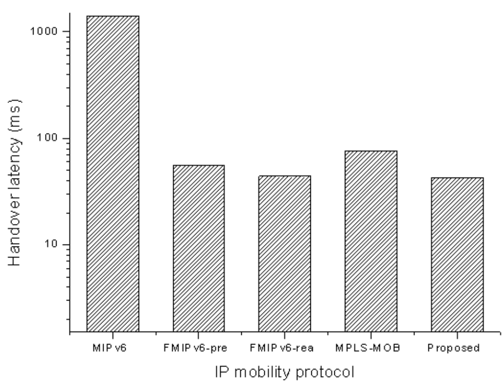


Fig. 3. Comparison of handover latency

Figure 3 compares the handover latency of the mobility protocols analyzed. The figure shows that MIPv6 has the highest handover latency due to the long registration delay including the movement detection delay and IP configuration delay unlike other schemes. We can see that the proposed scheme has the lowest handover latency than others. It is because the proposed scheme not only delivers signaling by L2.5 switching based on MPLS LSP, it also receives first packet from old HCA which has buffered packets destined for an MN after handover. The existing MPLS-MOB has higher handover latency than two modes of FMIPv6. In the MPLS-MOB, first packet addressed to the MN after handover is sent from CN. On the other hand, in the FMIPv6, that is sent from PAR or NAR according to reactive mode or predictive mode. The highs and lows in the handover latency are caused according to such difference is the reason. However, FMIPv6 has to perform additional process such as binding update to HA or CN in succession as part of handover procedure even though it receives packet from PAR or NAR after handover. From location update perspective, exclusive of first packet delivery, the FMIPv6 has higher latency than the MPLS-MOB. The FMIPv6-pre has higher latency than the FMIPv6-rea. In the FMIPv6-pre, the PAR, upon receiving FBU, stops forwarding packets for the MN. So, the handover latency in the FMIPv6-pre has been started since that time. It is earlier than in the FMIPv6-rea.

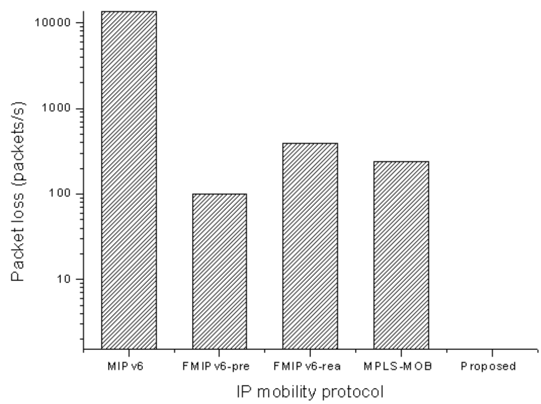


Fig. 4. Comparison of packet loss during handover

Figure 4 compares the number of lost packets during HO among the analyzed protocols. Because of long handover latency, MIPv6 suffers the largest amount of packet loss. FMIPv6-pre has less packet loss than FMIPv6-rea because it starts packet buffering while it is connecting to PAR. In FMIPv6, packets can be buffered after the PAR receives the FBU message and exchanges the HI and HACK with the NAR. The MPLS-MOB has less packet loss than FMIPv6-rea and has larger packet loss than FMIPv6-pre. Packet loss is lost packets until the MN receives first packet after handover. It can be associated with location update delay. Therefore, the FMIPv6-rea has more packet loss than the MPLS-MOB because it has higher latency than the MPLS-MOB for location update. Also, the FMIPv6-pre starts buffering packets destined for the MN when the MN is connecting to PAR. Therefore, the MPLS-MOB has larger packet loss than the FMIPv6-pre because it does not have packet buffering function. The proposed scheme adds the packet buffering function on the existing

MPLS-MOB. Therefore, the proposed scheme has the least packet loss among other schemes. The proposed scheme prevents lost packets during handover by using following functions: First, old HCA requests packet buffering to corresponding HCA while still being connected to the old HCA. Second, old HCA buffers in-flight packets which send to the MN before the corresponding HCA receives the packet buffering request from the old HCA. Third, MICS sends location update response to the old HCA for triggering to deliver the buffered packets from the old HCA to the new HCA. The location update response message is sent to the old and new HCA at the same time.

5 Conclusion

This paper proposes an improved MPLS-MOB scheme supporting small packet loss as well as low handover latency for seamless service in NGN. The proposed scheme reduce packet loss which may be occurred during handover by some process buffering and delivering the packets destined to the MN before and after connecting to new location according to the numerical results, we verified that the proposed scheme has lower handover latency and smaller packet loss than MIPv6, FMIPv6 and the existing MPLS-MOB.

In our future works, we will evaluate and compare the performance for the existing schemes such as [1], [2] and [4] as well as the proposed scheme by using simulators such as OPNET or NS-2.

Acknowledgments. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0023980).

References

1. Yu, M.J., Lee, J.M., Um, T.W., Ryu, W., Lee, B.S., Choi, S.G.: A New Mechanism for Seamless Mobility based on MPLS LSP in BcN. IEICE Trans. Commun. E91-B, 593–596 (2008)
2. Johnson, D., Perkins, C., Arkko, J.: Mobility Support in IPv6: RFC3775, IETF (2004)
3. Perkins, C.: IP Mobility Support for IPv4: RFC3344, IETF (2002)
4. Koodli, R.: Mobile IPv6 Fast Handovers: RFC5268, IETF (2008)
5. Yang, M.-j., Cheon, K.-Y., Park, A.-S., Choi, Y.-H., Kim, S.-H.: Seamless Handover using FMIPv6 with Effective Tunnel Management Scheme. In: GLOBECOM 2008, pp. 1–5 (2008)
6. Yeh, J.-H., Chen, J.-C., Agrawal, P.: Fast Intra-Network and Cross-Layer Handover (FINCH) for WiMAX and Mobile Internet. IEEE Transactions on Mobile Computing 8, 558–574 (2004)
7. Kong, K.-S., Lee, W., Han, Y.-H., Shin, M.-k.: Handover Latency Analysis of a Network-Based Localized Mobility Management Protocol. In: ICC 2008, pp. 5838–5843 (2008)
8. Faithi, H., Prasad, R., Chakraborty, S.: Mobility Management for VoIP in 3G Systems: Evaluation of Low-Latency Handoff Schemes. IEEE Wireless Communications 12, 96–104 (2005)

9. Vassiliou, V., Zinonos, Z.: An Analysis of the Handover Latency Components in Mobile IPv6. Journal of Internet Engineering 3, 230–240 (2009)
10. Yu, M.J., Kim, K.Y., Kim, H.S., Cho, K.S., Choi, S.G.: New Mechanism for Global Mobility Management Based on MPLS LSP in NGN. Communication Networking 119, 56–65 (2010)
11. Zhang, L.J., Pierre, S.: Evaluating the Performance of Fast Handover for Hierarchical MIPv6 in Cellular Networks. Journal of Networks 3, 36–43 (2008)
12. Park, B., Hwang, E., Latchman, H.: An Efficient Fast Neighbor Discovery (EFND) Scheme to Reduce Handover Latency in Mobile IPv6. In: ICAC 2006, pp. 1306–1310 (2006)

Strata: Wait-Free Synchronization with Efficient Memory Reclamation by Using Chronological Memory Allocation*

Eunhwan Shin, Inhyuk Kim, Junghan Kim, and Young Ik Eom

School of Information and Communication Eng., Sungkyunkwan University,
300 Cheoncheon-dong, Jangan-gu, Suwon, Gyeonggi-do 440-746, Korea
{comsky,kkojiband,gtgkjh,yieom}@ece.skku.ac.kr

Abstract. A locking is the typical mechanism to coordinate the race between multiple threads. But it downgrades the overall system performance due to the waiting time for the locked data to be unlocked. Wait-free synchronization is the one of the schemes to cope with the locking cost. The basic idea is making a replica of the shared data in order to manipulate it, and then applying the updated data. Due to the allocation of replicas without waiting, the most cost consuming step of wait-free synchronization is the reclamation of memory. This paper presents *strata*, a wait-free synchronization scheme with efficient memory reclamation. It allocates memory in the chronological order for efficient reclamation, and guarantees both update and read side wait-free in *O(1)* execution time.

Keywords: Wait-free synchronization, memory management.

1 Introduction

Taking out the full performance of microprocessor has been a key to the software developers recently. Living up to expectations, the performance of single-core processor has been dramatically improved, however, performance scaling stopped due to power consumption, wire delay, DRAM latency as well as limitation in instruction level parallelism (ILP). In recent times, the emergence of multi-core technology discovers a new potential to overcome the limitation of current processor's clock speed by exploiting the concurrency. This new paradigm carries out a significant number of problems according to use of shared data structures. To solve these problems, the efficient coordination of concurrent access has become the new challenge for software developers, as multithreaded application will become more common.

A locking is the typical mechanism to coordinate the race between multiple threads, which simply gives a lock to shared data to prevent the concurrent access of other

* This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (2010-0022570).

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 217–231, 2011.

threads, but it downgrades the overall performance due to the waiting time for the locked data to be unlocked. Wait-free synchronization is one of the schemes to cope with the cost of the locking and guarantees that every operation will be completed within the bounded number of steps. The wait-free scheme ensures lock-free access to the data. The basic idea is making a replica of the shared data in order to manipulate it, and applying the updated information to shared data. Thus, while the process of wait-free synchronization is in progress, the most cost consuming step is the allocation and reclamation of the memory for indirect access to shared data. If so, a problem on memory management scheme can critically affect to whole system performance. The modern garbage collection scheme is usually based on reference counting, where an object is responsible for keeping track of how many references are pointing to it. If the counter goes down to zero, the object is expected to release itself and allows its memory to be reclaimed. But, reading and updating a globally shared reference counter have large overhead especially in multi-core environment, since it requires frequent synchronization of cache line among cores [1][2].

This paper presents *strata*, a wait-free synchronization mechanism with memory reclamation by using chronological memory allocation. It performs batch memory reclamation that eliminates the cost of frequent update of the information for each object. Furthermore, it guarantees both update and read side wait-free and runs in *O(1)* time.

The rest of this paper is organized as follows: In Section 2, we discuss related work and briefly talk about *strata*. In Section 3, we present the design and implementation of *strata*. In Section 4, we evaluate and compare existing schemes with *strata*. In Section 5, we summarize the paper.

2 Related Work

This section reviews the lock-free synchronization schemes and the memory reclamation schemes; read copy update (RCU), hazard-pointer-based reclamation (HPBR), and lock-free reference counting (LFRC). We provide a short review of each scheme to help readers easily understand our idea.

2.1 Read Copy Update (RCU)

RCU [3][4] is a synchronization mechanism which allows lock-free read-only access to data structures that are being concurrently modified. It provides wait-free read with extremely low overhead. However, the update operation of RCU can be expensive, because they must wait until pre-existing readers finish their accesses.

The overall step of RCU is as follows; firstly, create a new structure, copy the data from the old structure into the new one, and save a pointer to the old structure; and then, secondly, modify the new copied structure, and update the global pointer to refer to the new structure; after then, finally, sleep until the operating system kernel determines that there are no readers left using the old structure.

The number of RCU core API is quite small; `rcu_read_lock()` marks an RCU-protected data and `rcu_read_unlock()` informs the reclaimer that the reader is exiting an RCU read-side critical section; `synchronize_rcu()` blocks until all pre-existing RCU read-side critical sections on all CPUs have completed; `rcu_assign_pointer()` assigns a new value to an RCU-protected pointer and `rcu_dereference_pointer()` fetches an RCU-protected pointer.

The key point underlying RCU is the concept of a grace period. The grace period is any time period during which each thread resides at least one quiescent state. A quiescent state means any statement that is not within an RCU read-side critical section. By introducing the new notion such as quiescent state and grace period, RCU shows high scalability for accessing and modifying read-mostly data structures. Though it still has the expense with update operation which does not underestimate when wait-free update is not guaranteed. We include some analytic comparisons with RCU to enumerate *strata*'s strengths with guaranteed both update and read-side of wait-free synchronization.

2.2 Hazard-Pointer-Based Reclamation (HPBR)

HPBR [5][6][7] is the lockless reclamation scheme and provides a locking mechanism for dynamically-allocated nodes. HPBR is associating a number of single-writer multireader shard pointers that called hazard pointers. A hazard pointer is an element used by a method that allows the memory allocated to the nodes of lock-free dynamic shared objects to be reclaimed. Each thread keeps a list of hazard pointers indicating which nodes the thread may access later. This list can only be written by the owning thread referencing that node. If no such pointers are found on the memory occupied by the node can be safely freed. Basically, hazard pointer is a mechanism for a thread to declare to other threads it has to watch for them. The paper analyzed a large number of lock-free data structures and found all requires only one or two hazard pointers per thread.

An algorithm using HPBR must recognize all hazardous references which references to shared nodes that may have been removed by other threads. Such references require hazard pointers. The algorithm sets a hazard pointer, and then checks for node removal; if the node has not been removed, then it may safely be referenced. While the hazard pointer references the node, HPBR's reclamation routine renounces from reclaiming it.

We include some analytic comparisons with HPBR to emphasize *strata*'s strengths with batch memory reclamation which does not need a special element to look out the operation of threads and the cost of frequent update.

2.3 Lock-Free Reference Counting (LFRC)

LFRC [7][8][9][10] is a lockless garbage-collection technique. Threads track for each node a count of the number of references to it held by other nodes. If a node's reference count reaches zero, the node has become inaccessible, and can be reused. Simple reference count mechanisms require frequent updates of reference variable. Whenever a reference is destroyed or overwritten, the reference count of the node it references is decremented, and whenever one is created or copied, the reference count of the node it reference is incremented. The main advantage of LFRC is that nodes are reclaimed as soon as they can no longer be referenced without any lock. Additionally, LFRC is also among the simplest forms of garbage-collection to implement.

LFRC scheme uses compare-and-swap (CAS) and fetch-and-add (FAA) in order to increment and decrement the reference counter. Using the single-address CAS to manipulate pointers and reference counters non-atomically is possible to make the order of operations tangled, according to the concurrent update of type and reference counter field of nodes, thus an obstacle of memory reclamation.

We will peer into *strata* in the remaining sections of paper. *Strata*'s core idea is based on using the concept of time-ordered memory allocation like chronologically piled up strata. In comparison with existing lock-free synchronization schemes, *strata* satisfies wait-free of both update and read side simultaneously resolving inefficiency of memory reclamation.

3 Design and Implementation

This section describes the design and implementation of *strata*, as well as the detailed explanation of *strata*'s core idea and motivation; first we briefly overview our system with the explanation about *strata* APIs. We go through the design and details of implementation in succession.

3.1 *Strata* API

Strata consists of read side APIs and update side APIs as shown in Table 1, and Figure 1 shows how each API communicates among the reader, updater, and reclaimer.

Table 1. The *Strata* runtime functions and their input parameters

***Strata* Runtime Interface**	
Read-Side	`strata_register`(Block header)
	`strata_unregister`(Block header, Reader index)
	`strata_reference`(Block header, Reader index)
Update-Side	`strata_get_data`(Block header)
	`strata_put_data`(Block header, Updated replica's pointer)

A reader thread should register and unregister itself at the beginning and end of the thread using `strata_register()`/`strata_unregister()`. It can access to data through `strata_reference()`. The updater uses `strata_get_data()` whenever it wants to modify shared data. It allocates a replica in order to avoid memory contention. To apply the updated information to shared data, the updater calls `strata_put_data()`. After this, the updated data is accessible from the readers. The pseudo code and the work process will be elucidated in the rest of this section.

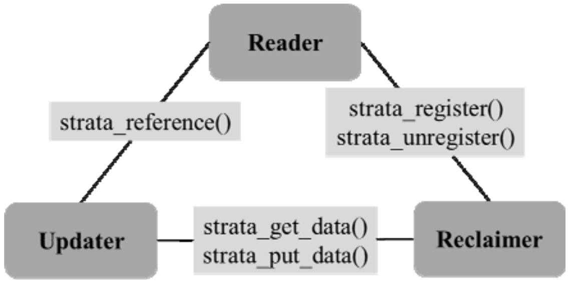


Fig. 1. How each API communicates among the reader, updater, and reclaimer

3.2 Batch Memory Reclamation

We illustrate the basic concept of *strata* in Figure 2. *Strata* uses a linked-list data structure, and the below contains a terminology of *strata* related terms which readers should know to catch the point of *strata*'s core idea with batch memory reclamation.

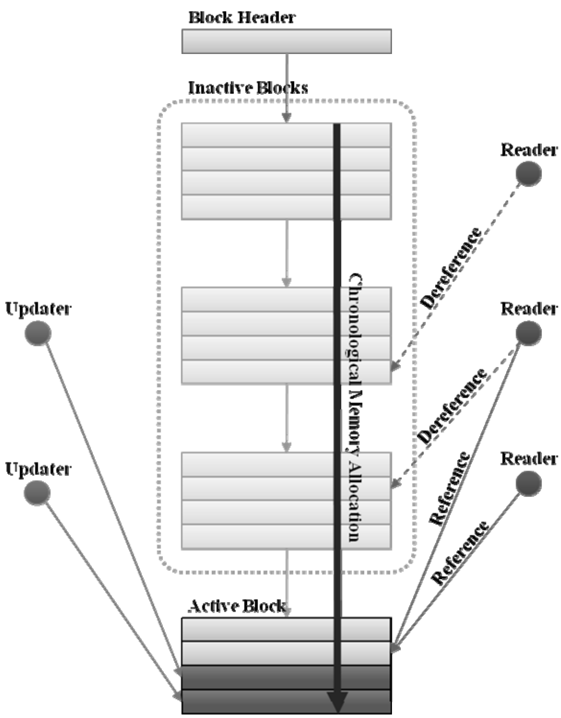


Fig. 2. Basic concept of *strata*

Chronological memory allocation. A memory allocation mechanism that assigns memory in recency order. For example, the top most block in Figure 2 is the oldest and the bottom most block is the newest. Each block is made up of the same number of objects that is predefined by user.

Active block/Inactive block(s). An active block is the group of objects most recently referenced by readers. It always located in the bottom most in order to chronological memory allocation. An inactive block is the rest of blocks except for the active block.

Reference/Dereference. A reference is the action that the reader requests a pointer of object to read the shared data. When the reader has done reference, it will release the pointer and stop to access to the shared data. We call this action as dereference which draws out dashed line in Figure 2.

The core idea of *strata*'s batch memory reclamation is discriminative treatment of the active block that only cares the reader that currently references the active block. *Strata* records which reader has referenced the object in active block, and when all registered readers have referenced the active block, performs batch memory reclamation from the top most block to the bottom most block except for the active block. For the batch memory reclamation, *strata* needs several information, so it maintains a block header per shared data structure. A block header is the data structure that contains information to handle the blocks and readers. The structure of block header is as shown in Figure 3.

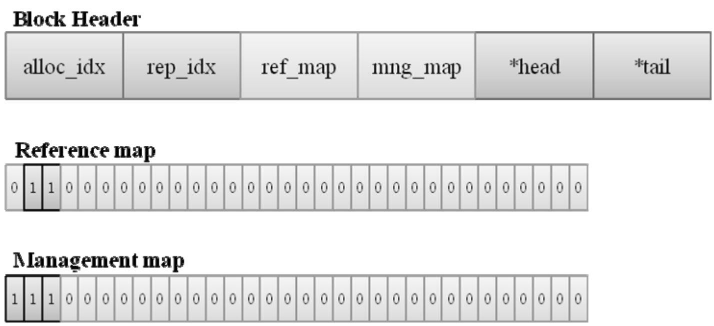


Fig. 3. Structure of block header

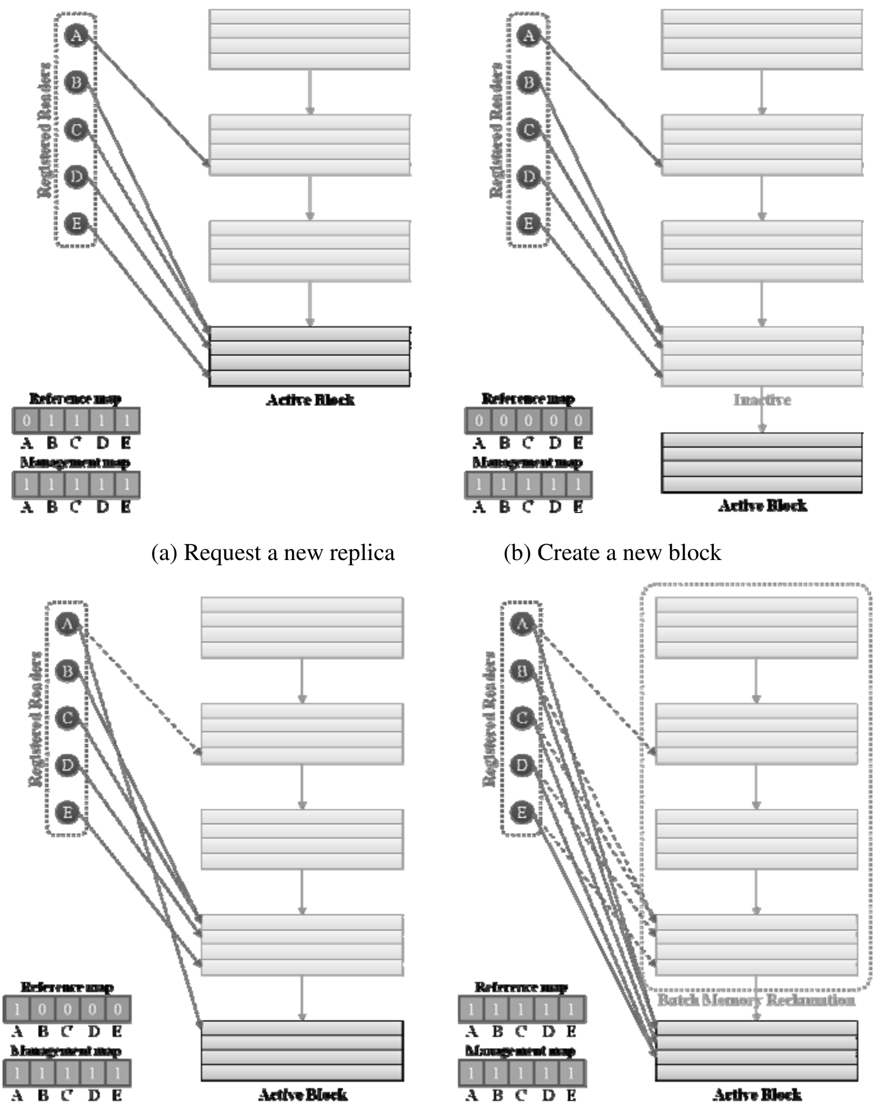
The components of block header are as follows; an allocation index, a replica index, a reference map, a management map, a head pointer and a tail pointer.

The allocation index is a value that simply stores the latest index of object in active block to recognize the number of object has reached to the maximum block size allowed. (e.g., the maximum block size in Figure 2 is 4). The replica index is a value that maintains the latest updated object's index in active block.

The management map is a bitmap that manages the register status of readers. To turn on (i.e., set as the value of 1) a single bit means a reader has registered to get permission to access the shared data. For example, the management map in the bottom of Figure 3 has 3 registered readers. The reference map which shown in the middle of Figure 3 also has a bitmap data structure that monitors which reader has referenced an object in active block. In Figure 3, there are total 3 registered readers and 2 of them have referenced the object in active block.

A head is a pointer to store the address of the top most block as well as a tail stores a pointer that points the active block. These two pointers provide the range of blocks for batch memory reclamation.

On the basis of above concepts, in Figure 4, we describe the basic scenario of batch memory reclamation to prove how *strata* works with chronologically ordered data structure step by step.



(a) Request a new replica (b) Create a new block

(c) Reference the new replica (d) Perform batch memory reclamation

Fig. 4. Basic Scenario of Batch Memory Reclamation

In Figure 4(a), we assume that there are 5 readers, which named A, B, C, D, E, registered in management map and 4 readers are now referencing the active block, so the bits of those readers in reference map have turned on excepting reader A. In Figure 4(b), an updater requests a new replica of the object for modification, so a new block is allocated because the size of block exceeded (i.e., Reader E is referencing the last object). At this point, *strata* resets the reference map, because the reference map manages the reference status of the only active block. Under the concept of chronological memory allocation, *strata* guarantees the reader will always access to the latest object in active block. Therefore, the use of *strata*, we don't need to waste extra space to keep the information of each object, and saving the cost of frequent update as well. In Figure 4(d), all of the registered readers reference the object in active block, and then *strata* performs the batch memory reclamation when the reference map is same as the management map.

The pseudo codes of strata API which mainly involves with batch memory allocation are below.

Table 2. Pseudo code of `strata_reference()`

```
00 program strata_reference()
01 Set block list points the top most block
02 Set a bit on reference map
03 IF reference map is same as management map THEN
04    WHILE until block list reaches before active block
05          Perform memory reclamation
06    END WHILE
07    Reset reference map
08    Reset head and tail
09 END IF
10 RETURN a pointer of the current replica
```

Table 2 shows the pseudo code of `strata_reference()`. The goal of this API is to get the pointer of current replica (i.e., the most recently updated replica) for reading. It accurately follows the mechanism of *strata* that we explained before. Lines 3-9 perform the batch memory reclamation within the scope of the head-to-tail.

Table 3. Pseudo code of `strata_get_data()`

```
program strata_get_data()
Increase allocation index
IF allocation index is greater than or equal to maximum block size
   Add new block
   Reset dereference map
   Reset allocation index
   RETURN a pointer of new replica in new block
ELSE
   RETURN a pointer of new replica in current block
END IF
```

The purpose of `strata_get_data()` shown in Table 3 is to obtain a pointer of the replica that recently updated to modify it. Lines 2-9 include the routine to check whether the allocation index has reached to maximum block size or not. If so, it will add a new block and return a pointer of newly created replica in the new block. Otherwise, it will simply return a pointer of newly created replica in a current block.

Table 4. Pseudo code of `strata_put_data()`

```
00 program strata_put_data()
01 IF existing replica index is greater than or equal to currently updated replica index
02    Ignore currently updated version of replica
03 ELSE
04    Set replica index to currently updated replica index
05    IF replica index has reached to maximum block size
06          Reset replica index
07    END IF
08 END IF
```

Table 4 shows the pseudo code of `strata_put_data()`. First, line 1 checks the currently updated replica index is meaningful or not. If not, it just ignores and doing nothing. If so, it sets the replica index to currently updated replica. Like the allocation index in `strata_reference()`, it checks the replica index has reached to maximum block size and resets the replica index.

3.3 Wait-Free Synchronization

The indispensable condition of wait-free synchronization is the guaranteed completion of thread's operation in constant time. The pseudo codes of `strata_register()` and `strata_unregister()` in Table 5 and Table 6 show how *strata* guarantees wait-free of both update and read side.

Table 5. Pseudo code of `strata_register()`

```
program strata_register()
Find the first zero bit in management map
Set a bit on management map
RETURN index of reader in management map
```

Table 6. Pseudo code of `strata_unregister()`

```
program strata_unregister()
Clear a bit on management map
Clear a bit on reference map
```

Those APIs above simply set or clear a bit on the bitmap, so *strata* can handle the status (e.g., the total number of readers that active/inactive) of the reader. Even though if the reader stays in inactive block indefinitely, *strata* still possible to guarantee wait-free, because all readers are under controlled. This concept is originally from Mathieu Desnoyers' User-level RCU (URCU). It is not automatically ensures, must guarantee by the software developers who use the API [11].

4 Evaluation

This section presents experimental performance comparison of *strata* and RCU as well as an analytic comparison of *strata* and well-known memory reclamation schemes. We select userspace RCU for comparison, because it includes RCU implementations that are based on primitives commonly available to user-level applications which like *strata* that implemented on the user-level. Also, we include the discussion about several implementation issues with *strata*. Then we review the strong point of *strata* comparing with other methods.

4.1 Performance Comparison with Userspace RCU

We considered about typically occurring race conditions among the multiple threads and the experiments were divided into three cases as follows. The overall experiments were performed on an Intel Core 2 Duo 2GHz dual-core processor with 4MB L2 caches and 3GB of physical memory.

Equivalent Race. The first case is that readers and writers are in the same condition without lean to one side. To make comparable race condition, we simply set the number of readers and writers to the same number that increases from 2 to 12 by even order. Then we accurately observed the trend of performance variation of both scheme.

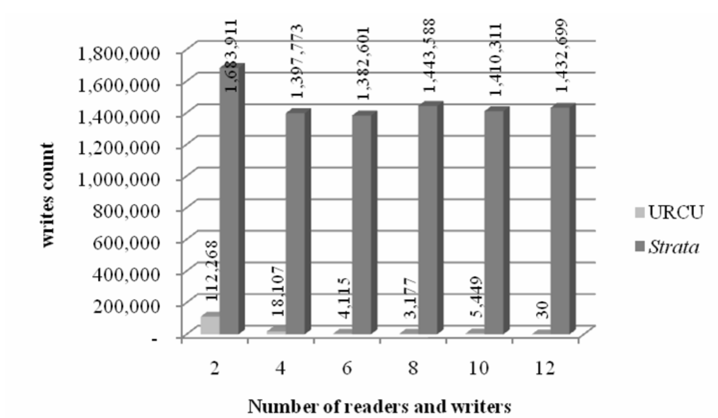


Fig. 5. Write Performance Evaluation

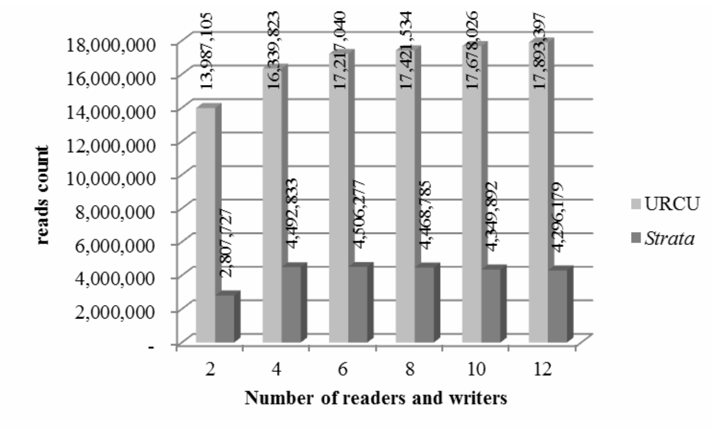


Fig. 6. Read Performance Evaluation

In Figure 5, the trend of write performance of userspace RCU does not look so satisfied and randomly bounced regard of the increment of number of readers and writers. On the contrary, *strata* shows at least 15 to 48,293 times higher write performance than userspace RCU. The userspace RCU basically uses lock-based write scheme that does not guarantee wait-free and lock-free access to shared data although *strata* guarantees wait-free synchronization by chronological memory allocation.

Figure 6 shows the read performance of each scheme. In Figure 6, userspace RCU has at least 4 to 5 times higher read performance than *strata*. The userspace RCU strongly focuses on the read-mostly data structure and introduces the concept of grace period to eliminate the waiting time of reader threads whereas *strata* focuses on the write-mostly data structure as well as the performance balance between read and write threads.

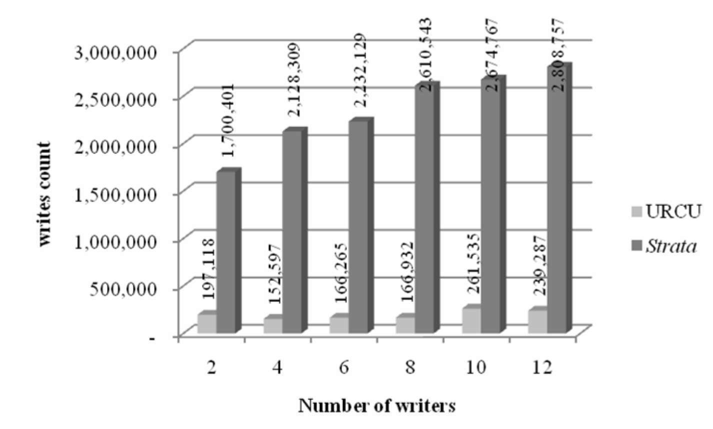


Fig. 7. Write-Intensive Write Performance Evaluation

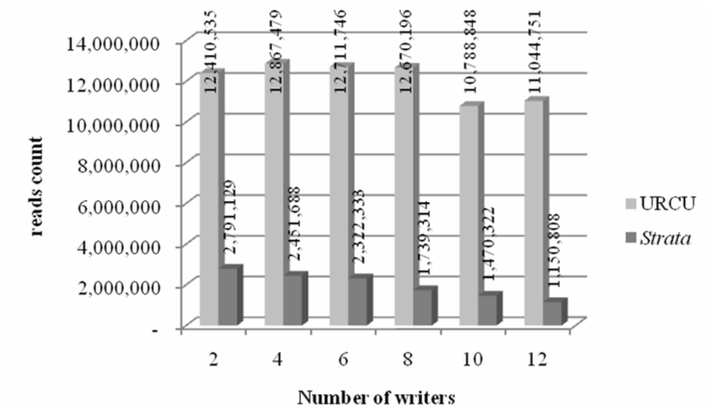


Fig. 8. Write-Intensive Read Performance Evaluation

Writer-Intensive Race. In the second case, we give limitations to the reader threads using a loopback routine to generate a delay time. We set the 100 times of looping and 2 readers with variation of writer number from 2 to 12 by even order. In effect, the reader threads consequently have a waiting time, so the performance of writer threads expects to improve relatively. Like the expected results, the write performance of userspace RCU shown in the Figure 7 is slightly increased, but still has bounced pattern. In contrast, in Figure 7, the write performance of *strata* has a linearly growing pattern that we expected, because the writer thread in *strata* does not need a waiting time to get the pointer of replica modifying the shared data.

Figure 8 shows the read performance of both schemes. All has decreased under the condition of small fixed number of reader threads according to the more competitive race between multiple writers.

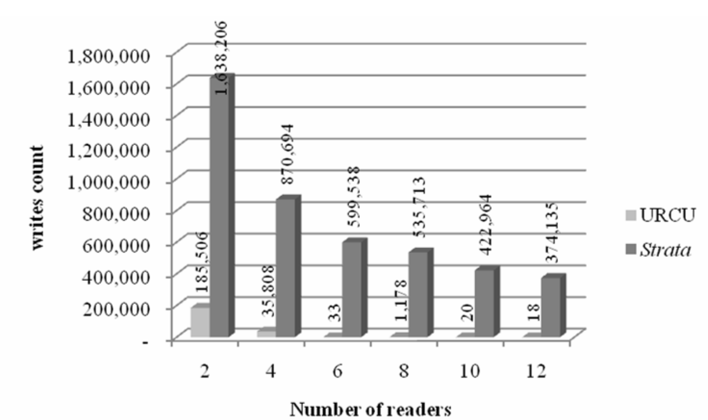


Fig. 9. Read-Intensive Write Performance Evaluation

Reader-Intensive Race. In the third case, we also give limitations to the writer threads using a loopback routine. We set the partially different condition that 100 times of looping and 2 writers with variation of reader number from 2 to 12 by even order. Ideally, we expect that *strata* maintains a constant write performance according to wait-free synchronization. However, the write performance of *strata* in the Figure 9 is rapidly decreased by increment in the number of reader threads. We find the cause of this problem in *strata* interface that it just implemented in prototype level and still has implementation issues to be fixed.

Figure 10 show the read performance of both schemes. *Strata* shows at most 30 percent of read performance comparing with userspace RCU. Nevertheless, there are some optimization techniques (e.g., cache miss optimization) to be applied, so read performance of *strata* is enough to improve in the future work.

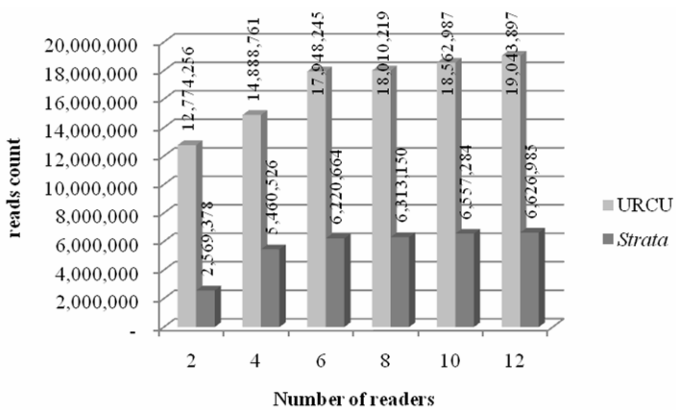


Fig. 10. Read-Intensive Read Performance Evaluation

As we can see in the overall results of performance comparison, the userspace RCU shows strength with the read-mostly data structure while *strata* shows strength with the write-mostly data structure.

In our motivation, *strata* originally focuses on the balance of performance between both update and read side, because there are lots of data structures that show the balance in the frequency of use (e.g., message queue in server or user interface). By using the concept of chronological memory allocation, *strata* almost removes the overhead with implementing non-blocking algorithm, but in our prototype implementation, there are still room for optimization.

Table 7. Comparison between execution time

	Read Side	**Update Side**
Userspace RCU	*O(1)*	*O(n)*
Strata	***O(1)***	***O(1)***

Table 7 shows the execution time of userspace RCU and *strata*. userspace RCU guarantees *O(1)* execution time with read side, but the execution time of update side is linearly increases by the number of updater. In comparison, *strata* has *O(1)* execution time both update and read side in order to guarantee wait-free and lock-free at the same time[3][4].

4.3 Analytic Comparison with Existing Memory Reclamation Schemes

Table 8 shows the result of analytic evaluation with *strata* and existing memory reclamation schemes. The first evaluation criterion is the number of additional element needs to use the scheme. HPBR requires a list of hazard pointers per object, and LFRC also demands a reference counter per object, but *strata* only uses one block header per shared data structure. The second is overhead with frequent update. As using lots of additional elements per object, we can easily know HPBR and LFRC spend higher cost for frequent update process than *strata*[5][6][7][8][9][10].

Table 8. Analytic Evaluation with Existing Memory Reclamation Schemes

	Number of Additional Element	**Overhead with Frequent Update**
HPBR	Per object	High
LFRC	Per object	High
Strata	**Only one**	**Low**

5 Conclusion

In this paper, we present *strata*, a methodology for efficient memory reclamation for wait-free synchronization. We show how *strata* uses a different memory management scheme that strictly follows the chronological order of memory allocation, and perform the basic scenario of batch memory reclamation using that scheme. Then we evaluate the performance of *strata* comparing with userspace RCU. *Strata* shows extremely higher write performance than userspace RCU according to wait-free update as well as reduces the cost of frequently update with simply defined data structure, and guarantee both update and read side wait-free with *O(1)* execution time.

References

1. Herlihy, M.: Wait-free synchronization. ACM Transactions on Programming Languages and Systems 13(1), 124–149 (1991)
2. Greenwald, M.: Non-blocking Synchronization and System Design. PhD thesis, Stanford University (1999)
3. McKenney, P.E., Appavoo, J., Kleen, A., Krieger, O., Russell, R., Sarma, D., Soni, M.: Read-Copy Update. In: Ottawa Linux Symposium (2001)
4. McKenney, P.E., Slingwine, J.D.: Read-Copy Update: Using Execution History to Solve Concurrency Problems. In: 10th IASTED Parallel and Distributed Computing and Systems, pp. 509–518 (1998)

5. Wikipedia, http://en.wikipedia.org/wiki/Hazard_pointer
6. Michael, M.M.: Hazard Pointers: Safe Memory Reclamation for Lock-Free Objects. IEEE Transactions on Parallel and Distributed Systems 15(6), 491–504 (2004)
7. Hart, T.E., McKenney, P.E., Brown, A.D.: Making Lockless Synchronization Fast: Perform-ance Implications of Memory Reclamation. In: 20th IEEE International Parallel and Distrib-uted Processing Symposium (2006)
8. Detlefs, D.L., Martin, P.A., Moir, M., Steele Jr., G.L.: Lock-Free Reference Counting. In: 20th IEEE ACM Symposium Principles of Distributed Computing, pp. 190–199 (2001)
9. Gidenstam, A., Papatriantafilou, M., Sundell, H., Tsigas, P.: Efficient and Reliable Lock-Free Memory Reclamation Based on Reference Counting. In: 8th International Symposium on Parallel Architectures, Algorithms and Networks, vol. 20(8), pp. 1173–1187 (2005)
10. Sundell, H.: Wait-Free Reference Counting and Memory Management. In: 19th IEEE International Parallel and Distributed Processing Symposium (2005)
11. Desnoyers, M., McKenney, P.E., Stern, A., Dagenais, M.R., Walpole, J.: User-Level Implementations of Read-Copy Update. IEEE Transactions on Parallel and Distributed Systems (2009) (to appear)

DDAT: Data Dependency Analysis Tool for Web Service Business Processes

Ivaylo Spassov[1], Valentin Pavlov[1], Dessislava Petrova-Antonova[2], and Sylvia Ilieva[3]

[1] Rila Solutions, Acad. G. Bonchev str., bl. 27, 1113 Sofia, Bulgaria
[2] FMI, Sofia University, 5, J. Bouchier str., 1164 Sofia, Bulgaria
[3] IICT, Bulgarian Academy of Sciences,
Acad. G. Bonchev str., bl. 25A, 1113 Sofia, Bulgaria
{ispasov,vpavlov}@rila.bg, d.petrova@fmi.uni-sofia.bg,
sylvia@acad.bg

Abstract. Service Oriented Architecture (SOA) provides a new generation of software architectures intended to integrate loosely coupled applications. In order to meet changing business goals, SOA systems incorporate multiple independent services supported by various providers. The complex structure of the resulting business processes can be defined in Business Process Execution Language for Web Services (WS-BPEL).

One of the main tasks of the quality assurance specialists during software design is testing the execution of different paths of the WS-BPEL process. In order to solve that task we propose an approach that augments the WS-BPEL process for test case generation by analyzing the conditional activities concerning given path of testing.

Keywords: design time testing, data dependency analysis, SOA, WS-BPEL.

1 Introduction

Service Oriented Architecture (SOA) provides a new generation of software architectures intended to incorporate loosely-coupled applications. A common implementation solution is based on web services delivered over the network. Often, their functionality is combined into more complex services (composite web services) interacting in the context of business processes described usually with Business Process Execution Language for Web Services (WS-BPEL) [1]. The business process is a "collection of activities performed by human users or software applications that together constitute the different steps to be completed to achieve a particular business objective" [2]. WS-BPEL defines a model and a grammar for describing the behavior of a business process based on interactions between the process and its partners. The WS-BPEL process defines how multiple service interactions with these partners are coordinated to achieve a business goal, as well as the state and the logic necessary for this coordination [3].

One of the main tasks of the quality assurance specialists during software design is testing the execution of different concurrent sequences of activities (paths) of the WS-BPEL process. The concurrent paths are possible due to conditional activities that

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 232–243, 2011.

cause appearance of branches in the execution of the WS-BPEL process. For example, let's consider an activity IF (C, A, B). If the condition C is satisfied, then the WS-BPEL process's execution follows the sequence of activities, starting with A. Otherwise, the sequence of activities, starting with B, will be executed. Thus, the WS-BPEL process can continue its execution on two different paths depending on the condition C.

In order to test the WS-BPEL process or a given path within, the quality assurance specialists have to identify all possible conditions of that path and the variables they depend on. Then, the quality assurance specialists have to set appropriate values of the variables so that the WS-BPEL process continues its execution on the desired path. When the WS-BPEL process is too complex (for example a process with branch nesting of more than three levels) this task may prove very intensive. However, all necessary data for solving this task are in the description of the business process that is represented by WS-BPEL, Web Service Description Language (WSDL) and XML Schema Definition (XSD) documents. For a given path of the WS-BPEL process the proposed approach finds all conditional activities along the path, specifies which variables affect those conditional activities and sets the variable values appropriately for the WS-BPEL process to follow the specified path. The approach is implemented in a tool called Data Dependency Analysis Tool (DDAT), which is one of the tools provided by testing framework named TASSA [4].

The main goal of the TASSA framework is to support the testing, validation and verification of both functional and nonfunctional behavior of service-based applications at design time as well as at runtime. Besides the DDAT the TASSA framework provides four other tools, which integrated together offer to developers and service integrators the complete environment for design time functional testing of service-based applications at WS-BPEL level. The *Isolation Tool* provides functionality needed to isolate the WS-BPEL under test from external dependencies. The *Injection Tool* injects delays, errors, etc. in the message exchange for a particular WS-BPEL activity. The *Value Generation Tool (VGT)* provides appropriate values that DDAT needed for the variables of the conditional activities. The *Test Case Generation Tool (TCGT)* generates test cases that traverse the WS-BPEL for all possible execution paths. DDAT has inputs from the user (via a Graphical User Interface (GUI) inside the integrated development environment (IDE) or from command line (via automated interface) and gives output to the User Interface (UI).

The rest of the paper is organized as follows. Section 2 surveys the current state of the art. Section 3 presents a formal description of the proposed approach for data dependency analysis of WS-BPEL processes. Section 4 focuses on the implementation of the approach in the software tool named DDAT. Section 5 shows experimental results from execution of the DDAT over a few simple BPEL processes. Finally, section 6 concludes the paper giving directions for future work.

2 Related Work

The WS-BPEL inherently brings a challenge for testing due to its specific syntax, dynamic binding during execution and the fact that it integrates web services implemented by various providers. This section presents a brief review of various techniques, methods and tools that meet this challenge. Each of them uses a specific intermediary model for generation of executable paths of the WS-BPEL process and test cases. The testing techniques that derive test data from a model, describing expecting behavior of the software under test, are called model-based [15]. Several formal verification methods using models, such Model checking, Control Flow Graphs (CFGs) and Petri nets (PNs), can be accepted as model-based testing approaches. The model-driven testing is also applicable to the WS-BPEL testing. It separates the testing logic from the actual test implementation [23]. UML 2.0 Testing Profile is an example for application of model-driven testing.

The approach presented in [16] relies on model-based test data generation for semantic web services. The pre-defined fault-models and Inputs, Outputs, Preconditions and. Effects (IOPE) information from semantic specification are used for test case generation. Automata, called Web Service Time Extended Finite State Machine (WS-TEFSM), and Intermediate Format (IF) are use in the approach proposed in [17] and [18] for generation of time test cases that exercise the time constraints in web service compositions. In [19] the BPEL processes are modeled as a Message Sequence Graph (MSG) and test cases are generated using this MSG.

Search-based test data generation techniques can be applied also to BPEL testing process [20], [21]. Scatter search that works with a population of solution is proposed in [22] as a test generation technique for BPEL compositions. Here, the BPEL composition is transformed to a state graph and test cases are generated according to transition coverage criterion.

In [5] authors propose graph-search based approach to WS-BPEL test case generation, which transforms the WS-BPEL process into an extension of CFG, traverses the graph and generates of test paths using a constraint solving method. The basis path testing that is presented in [6] starts with creating a CFG from which McCabe's cyclomatic complexity is computed. As a result, the number of basis paths is determined, which corresponds to the number of test cases needed for the flow. In [7] the authors propose a gray-box testing approach that has three key enablers: test-path exploration, trace analysis, and regression test selection. During WS-BPEL test-case generation the WS-BPEL process is transformed into CFG supporting fault-handling logic. The approach in [8] uses an extended CFG to represent a WS-BPEL program, and generates all the sequential test paths from it. Fault and event handling is presented in the graph together with WS-BPEL basic activities.

In [9] the authors use High-level Petri nets (HPNs) to model WS-BPEL web service composition. The approach proposed in [10] addresses the problem of checking and quantifying how much the actual behavior of a service, as recorded in message logs, conforms to the expected behavior as specified in a process model. The expected behavior is defined using the WS-BPEL. WS-BPEL process definitions are translated into PNs.

The approach presented in [11] extends UML2.0 activity diagram to describe the syntax and behaviors of WS-BPEL. The test coverage criteria of UML2.0 activity diagram includes following coverage criterions: Action, Transition, Basic path, and Sequence. Stream X-Machine is applied in [12] to automatically generate test cases for WS-BPEL process. In [13] Web Service Automata is proposed to model concurrency, fault propagation, and interruption features of WS-BPEL process. In [14] the model checking technique is applied to generate test case specifications for WS-BPEL compositions.

All of the above presented approaches are automated. Some of them, for example [13] and [14] rely on external components like SPIN (Simple ProMela INterpreter) and NuSMV (Symbolic Model Verifier) model checkers. Others as [7] and [6] implement fully the proposed approach.

Our research shows that the CFG transformation of the WS-BPEL process is used in the most of the approaches. This is due to the fact that such transformation allows the conventional graph algorithms for path searching to be applied, while the model-checking techniques use domain specific language to describe the model, which burden the testing study process. Compared to PNs, CFG allows easy coverage of the exception and fault handling WS-BPEL activities. That is why we have chosen the CFG as intermediate representation of the WS-BPEL process in our approach.

3 Formal Description of the Approach

This section presents a formal description of the approach for data dependency analysis of WS-BPEL processes.

Definition 1. Let *D* is a set of all variables of the WS-BPEL process, nevertheless of their type. Each process's variable can participate in the description of the conditions and each condition works only with process's variables and constants.

Definition 2. Let *F* is a set of all boolean functions over one or several variables of the WS-BPEL process. The functions of *F* are built from basic WS-BPEL operations that can be described with XML Path language (XPath) expressions.

Definition 3. Let $G = (V, E)$ is a control flow graph (CFG) that represents the WS-BPEL process, where *V* is a set of vertices and $E \subseteq \{(u, f \times v) \mid u, v \in V, f \in F\}$ is a set of edges, representing directed conditional binary relation between vertices.

The vertices of the CFG represent the activities of the WS-BPEL process and the edges correspond to the possible transitions from activity u to activity v. The realization in fact of the transition depends on its condition f.

Definition 4. Let $p = [v_1, v_2, ... v_n]$ is a path in the CFG, representing a sequence of vertices that correspond to a executable sequence of WS-BPEL activities.

Definition 5. Let $P \subseteq 2^V$ is a set of valid paths that can be traversed.

Definition 6. Let *cdr(p)* is a right sub-path of the path *p*. If $p = [v_1, v_2, ... v_n]$ then $cdr(p) = [v_2, .., v_n]$.

Lemma 1. If $u \in V$ and $e = (u, f \rightarrow v) \in E$ then $[u, v] \in P$.

Proof. According to Definition 3, the transition between *u* and *v* is possible because of presence of an edge *e*, *Lemma 2.* If $w \in V$, $e = (w, f \to v_1) \in E$ and $p = [v_1, v_2, ... v_n] \in P$ then $[w, v_1, v_2, ... v_n] \in P$.

Proof. According to Definition 3, the transition between w and v_1 is possible because of presence of an edge e. The transitions from v_1 to v_2 as well as from v_2 to v_3 and so on are possible because they are part from the valid path p. Therefore, all transitions between the consecutive activities on the path $[w, v_1, v_2, ... v_n]$ are possible, i.e. it can be traversed in fact during execution of the WS-BPEL process and is valid.

Corollary of Lemma 1. For each path $p = [v_1, v_2, ... v_n]$ the following implication is correct:

$$cdr(p) \in P \wedge \exists e = (v_1, f \to v_2) \in E \Rightarrow p \in P$$

or in other words, if the right sub-path of p is valid and there is an edge between the first two vertices of p in the CFG, then p is a valid path.

The main task of the approach is to find all conditional activities that belong to a given path of the WS-BPEL process and for each condition to find all variables that participate in it. The pseudo-code of the algorithm that implements that task is presented bellow.

```
FUNCTION FIND(p)
1.        Init(D,G)
2.        d = 0, f =0
3.        FOR ∀a ∈ p
4.          FOR ∀e ∈ E ∧ e.u = a
5.            IF match(e.v, cdr(p)) DO
6.              f = f ∪ e.f
7.              d = d ∪ dex(e, f)
8.            ELSE RETURN p is invalid
9.          END FOR
10.       END FOR
11.       RETURN d, f
```

On the first step the WS-BPEL, WSDL and XSD documents related to the given business process are analyzed. As a result, the set D of all variables and the graph *G* are initialized (line 1). The second step (line 2) initializes the output variables. Then in step 3, for each activity that belongs to the path *p*, step 4 is executed, where for each possible branch after activity *a*, step 5 is executed. The step 5 uses the function match in order to verify whether the path *[e.v,cdr(p)]* is valid. For that purpose, the corollary of Lemma 1 is recursively applied. If the path *[e.v,cdr(p)]* is valid, then the condition *e.f* has to be satisfied in order for the process to continue execution on the path p. Hence, *e.f* will be added to the set of conditions *f* (step 6) and the process's variables *dex(e,f)* that participate in it are added to the set of variables *d* (step 7). If there is not a condition that leads to execution of the path *cdr(p)*, the path *p* is invalid (step 8).

The function *dex* on the step 7 extracts the variables that belong to a given condition. In fact, it is a parser that interprets the basic operations of the WS-BPEL library. The function *dex* is a key point of the implementation of the algorithm because the WS-BPEL library is fairly large. It is realized by standard methods for implementation of interpreters. The examination of the grammar shows that the LR(1) parser is particularly well-suited. Java Compiler Compiler (JavaCC) is used for creation of LR(1) parsers and the grammar of the XPath expressions from the standard WS-BPEL library is represented. The final code generator implements the function dex in case of given XPath expression, identifies the variables and return them as a set.

Figure 1 presents a block diagram of the algorithm.

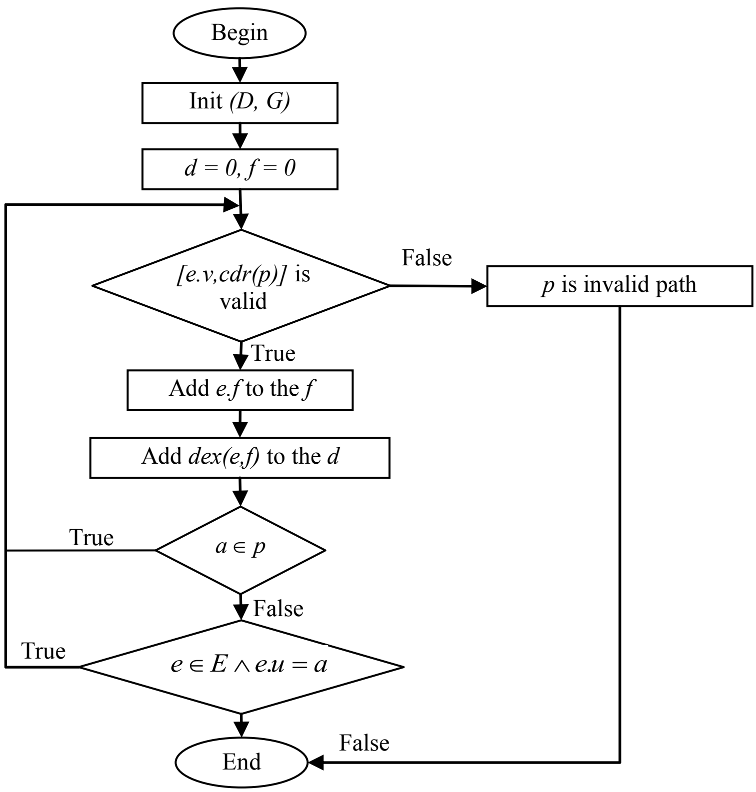


Fig. 1. A Block diagram of the algorithm for data dependency analysis

4 Implementation of the Approach

This section presents the implementation of the approach as a software tool called Data Dependency Analysis Tool.

DDAT consists of the following components: Data Dependency Analysis Web Service (DDAWS), Graphical User Interface (GUI) and Command Line Interface (CLI). The architecture of DDAT is shown on Figure 2.

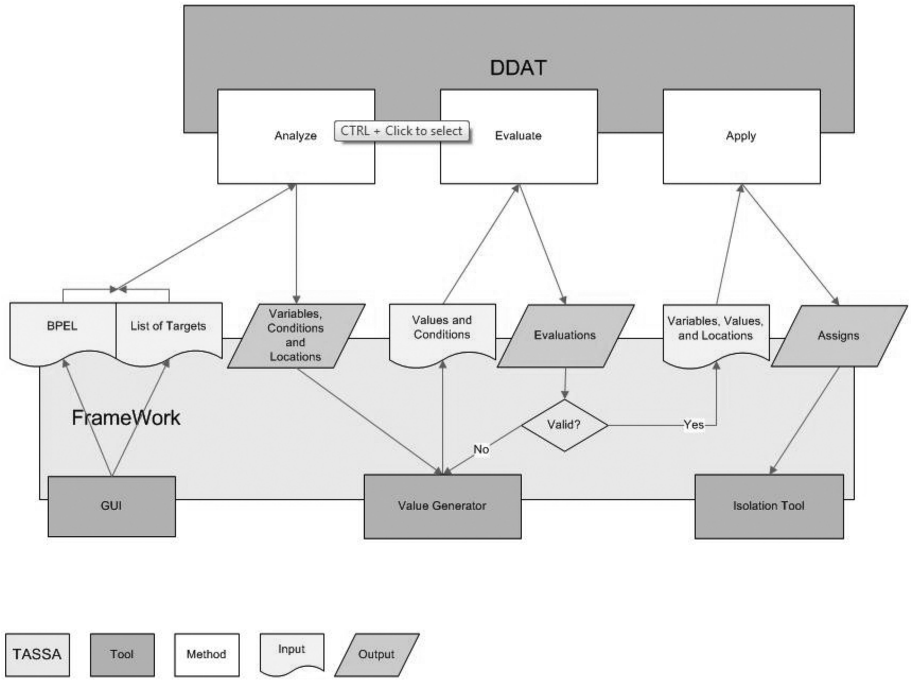


Fig. 2. Architecture of DDAT

The main functionality of DDAT described in section 2 is implemented as operation of the DDAWS, called Analyze. The operation Analyze returns a list of conditions and for each condition a list of variables, belonging to that condition, and location where the variables have to be injected. It receives as input a WS-BPEL process and array of unique identifiers of activities, describing the path that the WS-BPEL process needs to follow. The XPath expression that represents given activity of the WS-BPEL process is used as unique identifier. The output of the operation Analyze is represented in text mode.

The DDAT interacts with two other operations provided by TASSA framework:

- Evaluate – with given values of the variables, check whether corresponding conditions are performed so the WS-BPEL process execution follows the preliminarily defined path. This operation uses API’s that are already implemented in the TASSA framework.

The input of the operation Evaluate is an array of conditions, together with specific variable values. It returns an array of Boolean values, which indicates if their corresponding conditions are appropriately satisfied so the WS-BPEL process to continue execution on the specified path.

- Apply – with given values of the variables, belonging to the corresponding conditions and specific injection locations, transforms the WS-BPEL process in such a way so that the variables obtain the necessary values before condition checking.

The input for the operation Apply is the WS-BPEL representation of the process, the specific injection locations, and the specific values to inject. It calls Isolation tool of the TASSA framework by which Assign activities are inserted before the corresponding conditions in order to set the variable values. The output of the operation Apply is a transformed WS-BPEL description of the process, whose execution will cause the process to continue on the desired path.

The GUI allows the integration of other TASSA tools. It allows users to define the path that the WS-BPEL process has to pass. The user receives the CFG representation of the WS-BPEL process and selects the desired target activities. Then the same GUI is used to visualize the results of the DDAWS, and later by the Evaluate and Apply completes the data analysis task, as described in previous paragraph.

The CLI works without user intervention. It is used mainly from the other tools of the TASSA framework and presents the results from execution of the operation Analyze.

5 Experimental Results

The proposed approach has been validated through application on a few simple BPEL processes. This section presents two of them, called Travel Reservations and Order Music.

The Travel Reservations WS-BPEL process is presented on Figure 3.

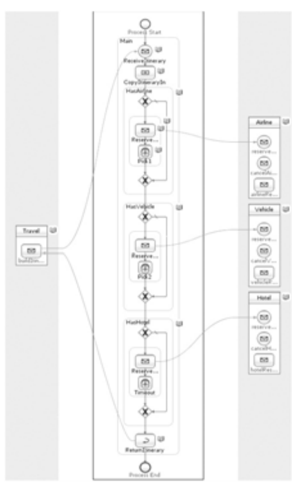


Fig. 3. Travel Reservation Process

The Travel Reservations WS-BPEL process consists of five tasks: receiving of request for reservation (receive activity), creating an airplane reservation (invoke activity), reserving a vehicle (invoke activity), reserving a hotel (invoke activity) and reply of the request (reply activity). Also the process has three conditional activities.

The first conditional activity checks if there is airplane registration for the customer. If there is not, then the WS-BPEL process invokes the Airline Registration Web Service. Otherwise, the process skips that invoke.

The second conditional activity checks if there is a registered vehicle for the customer. If there is no registered vehicle, then the WS-BPEL process invokes the Vehicle Registration Web Service. Otherwise, the WS is skipped.

The third conditional activity checks if there is a hotel reservation for the customer. If there is no registration then the WS-BPEL process invokes the Hotel Reservation Web Service. Otherwise the process continues without this WS call.

Consider the following example. Suppose that the invocation of the Airline Reservation Web Service and the Hotel reservation Web Service need to be tested. Therefore, the WS-BPEL process has to satisfy the first and the third conditions (Figure 4):

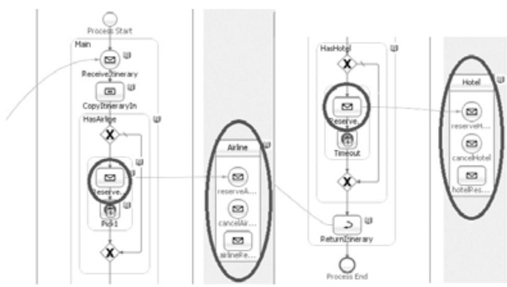


Fig. 4. Choosing target activities

The location of the invoke activities of the WS-BPEL process are identified by XPath. Then the DDAWS is called through the Command-line interface (CLI) (Figure 5):

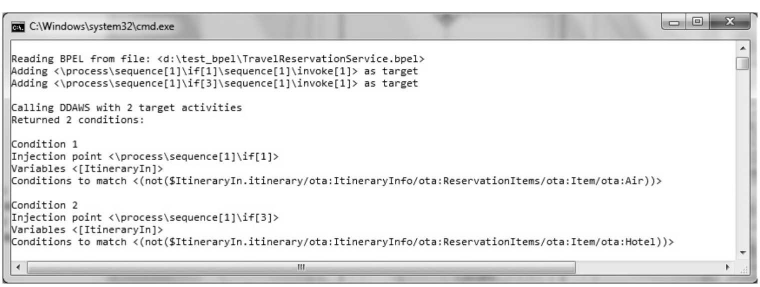


Fig. 5. Example call

The returned result set consists of two conditions. Each of them is supplemented with its place in the process, and the variables that determine its outcome. This information is then used to call another tool of the TASSA framework, in order to modify the WS-BPEL.

Indeed if the correct values (satisfying the cited conditions) of the variables are injected at the correct place (just before entering the first conditional activity), the service will be invoked.

The Order Music process is a sample developed by TASSA team. This example is constructed using majority of BPEL elements and three partner web services. Two of them are external. They are used for e-mail validation and checking of music tracks. The third one provides offers to the customers and is implemented by the TASSA team.

The Order Music process executes simple order of music tracks. The business process consists of four main steps. First, the buyer (customer) provides his/her personal information like name and e-mail address and the music artist or album he/she is interested in. Next, a partner web service verifies the e-mail address provided by the customer. If the e-mail address is wrong, the business processes ends with informative message. If the e-mail is correct, the customer's order is processed. Then, the process invokes another partner web service that provides information about the available music tracks of the chosen artist or album. Finally, a third partner web service calculates the prices of the music tracks and makes an offer to the customer. If the order is less than 100 EUR the calculated price is shown to the customer and he/she can buy the music tracks of his/her favorite artist or album. If the order is greater than 100 EUR a message is send to the customer that the company could offer him a better price for that quantity.

The complexity of WS-BPEL processes that have been used for verification of the proposed approach is presented in Table 1. In the first column is the name of the business process. Column called "Number of Activities" presents the total number of activities in the specified BPEL. In the third column is the total number of conditions and in the forth on is the maximum level of nested conditions that is used in the BPEL. "Max Variable Dependability" shows the maximum number of variables that affect these conditions. Analysis of more complicated BPEL is in progress although the results from the simple ones give a prove of concept, too.

Table 1. List of Analyzed BPELs

BPEL Process	Number of Activities	Number of Conditional Activities	Levels of Nesting	Max Variables Dependability
Travel Reservation	32	3	1	3
Order Music	21	3	2	3
ATM	54	4	3	4
MetaSearch	76	9	4	9
ASTROBookStore	232	23	6	23

6 Conclusion

One of our current goals is to test the execution of different paths of the WS-BPEL processes. When the WS-BPEL processes are too complex, this task may turn out very intensive. In order to meet this challenge, we propose an approach that finds all conditional activities for a given path of the WS-BPEL process, specifies which variables affect these conditional activities and sets the variable values appropriately for the WS-BPEL process to follow the specified path. The approach is implemented in a tool called DDAT, which consists of three components: 1) a web service, implementing the proposed approach, 2) a graphical user interface and 3) a command line interface. The validity of the approach and the DDAT are demonstrated through experiments.

Our future work will be concentrated on evaluation of the DDAT according to specified characteristics like effectiveness and dealing with WS-BPEL processes with different level of complexity. Its real value will be demonstrated in the further integration with other TASSA framework tools. Extending the usability of the GUI (Graphical User Interface) is also part of the planned activities. A certain limitation for the time being is that the tool only recognizes conditions expressed in XPath language as prescribed by the WS-WS-BPEL standard. However, that limitation is not principal and in the future we will try to support also conditions expressed e.g. in Java. Such advanced features are used in IBM's version of WS-BPEL.

Acknowledgements. The authors acknowledge the financial support by the National Scientific Fund, BMEY, grant agreement number DO02-182.

References

1. Petrova-Antonova, D., Krasteva, I., Ilieva, S.: Approaches Facilitating WS-BPEL Testing. In: 17th Conference on European Systems & Software Process Improvement and Innovation (EuroSPI2 2010), Grenoble Institute of Technology, France, pp. 5.1–5.17 (2010)
2. Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web Services: Concepts, Architectures and Applications (Data-Centric Systems and Applications). Springer, Berlin (2004)
3. Web Services Business Process Execution Standard Version 2.0, `http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html`
4. Pavlov, V., Borisov, B., Ilieva, S., Petrova-Antonova, D.: Framework for Testing Service Compositions. In: 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, Timisoara, Romania (2010)
5. Yuan, Y., Li, Z., Sun, W.: A graph-search based approach to BPEL4WS test generation. In: International Conference on Software Engineering Advances (2006)
6. Lertphumpanya, T., Senivongse, T.: Basis path test suite and testing process for WS-BPEL. WSEAS Transactions on Computers 7(5), 483–496 (2008)
7. Li, Z.J., Tan, H.F., Liu, H.H., Zhu, J., Mitsumori, N.M.: Business-process-driven gray-box SOA testing. IBM Systems Journal 47, 457–472 (2008)
8. Yan, J., Li, Z., Yuan, Y., Sun, W., Zhang, J.: Bpel4ws unit testing: Test case generation using a concurrent path analysis approach. In: Proc. of ISSRE, pp. 75–84. IEEE Computer Society, Los Alamitos (2006)

9. Dong, W.L., Yu, H., Zhang, Y.B.: Testing BPEL-based web service composition using high-level Petri Nets. In: Proceedings – IEEE International Enterprise Distributed Object Computing Workshop (2006)
10. Van Der Aalst, W.M.P., Dumas, M., Ouyang, C., Rozinat, A., Verbeek, E.: Conformance Checking of Service Behavior. ACM Transactions on Internet Technology 8(3), Article 13 (2008)
11. Zhang, G., Rong, M., Zhang, J.: A business process of web services testing method based on UML2.0 activity diagram. In: Proceedings - Workshop on Intelligent Information Technology Application, IITA 2007, pp. 59–60 (2007)
12. Ma, C., Wu, J., Zhang, T., Zhang, Y., Cai, X.: Automatic Test Case Generation for BPEL Using Stream X-Machine. International Journal of u- and e-Service, Science and Technology, 27–36 (2008)
13. Zheng, Y., Zhou, J., Krause, P.: An Automatic Test Case Generation Framework for Web Services. Journal of Software 2(3), 64–77 (2007)
14. García-Fanjul, J., Tuya, J., de la Riva, C.: Generating Test Cases Specifications for BPEL Compositions of Web Services Using SPIN. In: International Workshop on Web Services Modelling and Testing, pp. 83–94 (2006)
15. Bozkurt, M., Harman, M., Hassoun, Y.: TestingWeb Services: A Survey. Technical Report TR-10-01, Department of Computer Science, King's College London (2010)
16. Paradkar, A., Sinha, A., Williams, C., Johnson, R., Outterson, S., Shriver, C., Liang, C.: Automated functional conformance test generation for semantic web services. In: IEEE International Conference on Web Services, Salt Lake City, UT, USA, pp. 110–117 (2007)
17. Lallali, M., Zaidi, F., Cavalli, A.: Timed modeling of web services composition for automatic testing. In: IEEE International Conference on Signal-Image Technologies and Internet-Based System, Shanghai, China, pp. 417–426 (2007)
18. Lallali, M., Zaidi, F., Cavalli, A., Hwang, I.: Automatic timed test case generation for web services composition. In: 6th European Conference on Web Services, Dublin, Ireland, pp. 53–62 (2008)
19. Hou, S.S., Zhang, L., Lan, Q., Mei, H., Sun, J.S.: Generating effective test sequences for BPEL testing. In: 9th International Conference on Quality Software, Jeju, Korea (2009)
20. Ali, S., Briand, L.C., Hemmati, H., Panesar-Walawege, K.R.: A systematic review of the application and empirical investigation of search-based test-case generation. IEEE Transactions on Software Engineering 36(6), 742–762 (2010)
21. McMinn, P.: Search-based software test data generation: A survey. Software Testing, Verification & Reliability (STVR) 14(2), 105–156 (2004)
22. Blanco, R., García-Fanjul, J., Tuya, J.: A first approach to test case generation for BPEL compositions of web services using scatter search. In: IEEE International Conference on Software Testing, Verification, and Validation Workshops, Denver, CO, USA, pp. 131–140 (2009)
23. Model-driven Testing, `https://www.research.ibm.com/haifa/projects/verification/mdt/index.html` (last accessed April 6, 2011)

Towards a Quality Model for Semantic Technologies

Filip Radulovic[1] and Raúl García-Castro[2]

Ontology Engineering Group,
[1] Departamento de Inteligencia Artificial
[2] Departamento de Lenguajes y Sistemas Informáticos e Ingeniería Software
Facultad de Informática, Universidad Politécnica de Madrid, Spain
fradulovic@delicias.dia.fi.upm.es, rgarcia@fi.upm.es

Abstract. Semantic technologies have become widely adopted in the last years. However, in order to correctly evaluate them we need to ground evaluations in a common quality model. This paper presents some first steps towards the definition of such quality model for semantic technologies. First, some well-known software quality models are described, together with methods for extending them. Afterwards, a quality model for semantic technologies is defined by extending the ISO 9126 quality model.

1 Introduction

Software quality is acknowledged as a main need across domains (e.g., security, health) and technologies (e.g., operating systems, databases) and, in order to obtain high-quality software products, during the software development process the specification and evaluation of quality is of crucial importance [1].

One important component in software evaluation are software quality models, since they provide the basis for software evaluation and give a better insight of the software characteristics that influence its quality. Furthermore, quality models also ensure a consistent terminology for software product quality and provide guidance for its measurement.

In recent years, semantic technologies have started to gain importance and, as the field becomes more and more popular, the number of these technologies is increasing exponentially. Just as with any other software product, the quality of semantic technologies is an important concern. Multiple evaluations of semantic technologies have been performed, from general evaluation frameworks [2] to tool-specific evaluations [3,4] and even characteristic-specific evaluations [5]. However, the problem is that there is no consistent terminology for describing the quality of semantic technologies and it is difficult to compare them because of differences in the meaning of the evaluation characteristics used. Also, existing software quality models do not provide specification of quality characteristics that are specific to semantic technologies.

This paper describes a first step to build a quality model for semantic technologies, to provide a consistent framework to support the evaluation of such

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 244–256, 2011.

technologies, that extends the ISO 9126 quality model. To build this quality model, we have used a bottom-up approach, starting from real semantic technology evaluations and extracting from them the elements of the model.

This paper is structured as follows. Section 2 gives an overview of existing software quality models. Section 3 describes already defined methods for extending software quality models. Sections 4, 5 and 6 describe how we have defined the quality model . Finally, Section 7 draws some conclusions and includes ideas for future work.

2 Review of Software Quality Models

Quality in general is a complex and multifaceted concept that can be described from different approaches depending on whether the focus is in the concept of quality, the product, the user of the product, how the product was manufactured, or the value the product provides [6].

The way we define software quality depends on the approach that we take [7]; software quality means different things to different people and therefore defining and measuring quality will depend on the viewpoint. Similarly, choosing one software quality model or another will also depend on the intended users and uses of such model in concrete evaluations.

Next, this section describes some well-known software quality models and identifies their elements.

2.1 McCall's Model

McCall's software quality model is represented as a hierarchy of *factors, criteria* and *metrics* [8]. Factors are at the highest level in the hierarchy and represent the characteristics of the software product. Criteria are the middle layer and are considered to be the attributes of the factors, so that for every factor a set of criteria is defined. At the bottom level, metrics provide measures for software attributes.

McCall's model predefines a set of eleven software quality factors that are classified according to the product life cycle in three different groups:

– Product transition: portability, reusability, and interoperability
– Product revision: maintainability, flexibility, and testability
– Product operations: correctness, reliability, efficiency, integrity, and usability

McCall's quality model also gives the relationships between quality factors and metrics in the form of linear equations based on regression analyses. This is considered one of the major contributions of this model; however, the omission of the functionality aspect is regarded as the main lack [1].

2.2 Boehm's Model

Like McCall's model, Boehm's one has a hierarchical structure. It consists of twenty four quality characteristics divided into three levels [9]. It also gives a set of metrics and, while McCall's model is more related to the product view, Boehm's model includes users' needs.

2.3 ISO 9126's Model

The International Organization for Standardization (ISO) identified the need for a unique and complete software quality standard and, therefore, produced the ISO 9126 standard for software quality [10].

The ISO 9126 standard defines three types of quality: internal quality, external quality, and quality in use.

Six main software quality characteristics for external and internal quality are specified: Functionality, Reliability, Usability, Efficiency, Maintainability, and Portability, which are further decomposed into sub-characteristics (see Fig.1) that are manifested externally when the software is used, and are the result of internal software attributes [10]. The standard also provides the internal and external measures for sub-characteristics (see Table 1 for an example for Accuracy and Fault tolerance).

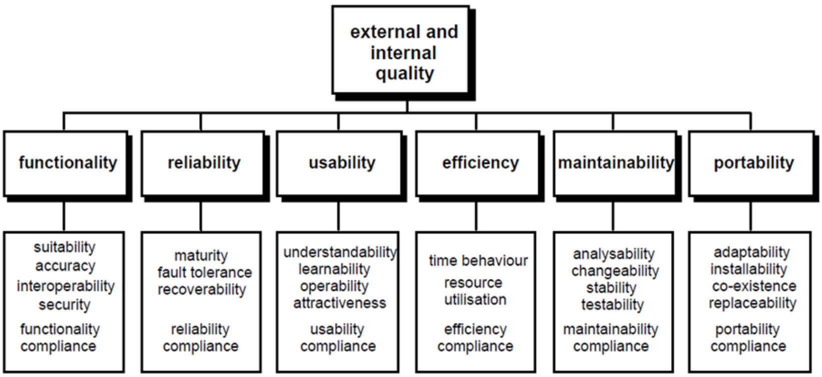


Fig. 1. ISO 9126 internal and external quality characteristics and sub-characteristics

Regarding quality in use, the model proposes four characteristics: Effectiveness, Productivity, Safety, and Satisfaction.

The ISO 9126 standard gives the complete view of software quality with evaluation criteria and precise definitions for all software characteristics and sub-characteristics. Some authors also suggests that according to the nature of the product itself some new sub-characteristics can be added, the definitions of existing ones can be changed, or some sub-characteristics can be eliminated from the model [11].

2.4 SQuaRE's Model

Although the ISO 9126 standard has been accepted and used successfully, some problems and issues for its further use and improvement have been identified. They eventually arise mainly because of advances in technologies and changes of users needs. As pointed out by Azuma [12], the main problems were due to issues on metrics and lack of a quality requirement standard.

Table 1. ISO 9126 internal and external measures for Accuracy and Fault tolerance

Quality Characteristics	Quality Sub-Characteristics	External Measures	Internal Measures
Functionality	Accuracy	Computational accuracy	Computational accuracy
		Precision	Precision
Reliability	Fault tolerance	Failure avoidance	Failure avoidance
		Incorrect operation avoidance	Failure avoidance
		Breakdown avoidance	

In order to address those issues, the existing standard is being redesigned and has been named SQuaRE. By the time of writing this paper, the parts of the SQuaRE standard related to the quality model and evaluations are still under development (ISO 25010, ISO 25040) and their final versions will be published in 2011.

As a summary of this section, Table 2 presents the elements of the described quality models. In our work, we have decided to adopt terminology from ISO 9126 standard.

Table 2. Elements of the described quality models

Structure/Model	McCall	Boehm	ISO 9126
First level	Factor	High level characteristic	Characteristic
Second level	Criteria	Primitive characteristic	Sub-characteristic
Third level	Metrics	Metrics	Measures
Relationships between entities	Factor-Metric	/	Measure-Measure

3 Approaches for Extending Software Quality Models

One existing software quality model (such as the ISO 9126) can be a good starting point for building a suitable quality model for a specific software product or domain. Its already defined characteristics and sub-characteristics should be considered, and some of them could be excluded or defined in a different manner, according to the nature of the domain. Also, the model can be extended by introducing new sub-characteristics if needed.

Software quality model extensions can be performed following two main approaches [13]:

- A **top-down** approach that starts from the quality characteristics and continues towards the quality measures.
- A **bottom-up** approach that starts from the quality measures and defines the quality sub-characteristics that are related to each specific measure.

In their work, Franch and Carvallo proposed a method based on a top-down approach for customizing the ISO 9126 quality model [14]. After defining and analyzing the domain, the method proposes six steps:

1. *Determining quality sub-characteristics.* In the first step, according to the domain, some new quality sub-characteristics are added while other are excluded or their definitions are changed.
2. *Defining a hierarchy of sub-characteristics.* If it is needed, sub-characteristics are further decomposed according to some criteria.
3. *Decomposing sub-characteristics into attributes.* In this step abstract sub-characteristics are decomposed into more concrete concepts which refer to some particular software attribute (i.e., observable feature).
4. *Decomposing derived attributes into basic ones.* Attributes that are not directly measurable are further decomposed into basic ones.
5. *Stating relationships between quality entities.* Relationships between quality entities are explicitly defined. Three possible types of relationships are identified:
   - *Collaboration.* When increasing the value of one entity implies increasing of the value of another entity.
   - *Damage.* When increasing the value of one entity implies decreasing the value of another entity.
   - *Dependency.* When some values of one entity require that another entity fulfills some conditions.
6. *Determining metrics for attributes.* To be able to compare and evaluate quality, it is necessary to define metrics for all attributes in the model.

In building their quality model for B2B applications, Behkamal et al. proposed a method to customize the ISO 9126 quality model in five steps [1]. The main difference with the previous method is that in Behkamal's approach the quality characteristics are ranked by experts; the experts should provide weights for all quality characteristics and sub-characteristics, and these weights are later used to establish their importance. Besides, Behkamal's approach does not contemplate defining relationships between quality entities.

These previous approaches follow the top-down approach, and we have not found any example of a bottom-up approach in the literature. Because of this, and because of the existence of plenty of evaluations in our domain (i.e., semantic technologies), we have decided to follow a bottom-up approach for extending the ISO 9126 quality model to cover semantic technologies. Evaluation results are used as the starting point, from which the quality measures, sub-characteristics and characteristics are specified.

Some authors have proposed software quality models for various types of applications: B2B [1], mail servers [15], web-based applications [16], e-learning systems [17], model-driven web engineering methodologies [18], and ERP systems [11]. All those authors have used the ISO 9126 standard as the basis software quality model, and have extended it to fit their particular domain.

Furthermore, some authors proposed introducing quality models for various steps of the product development, like in the case of web engineering [19]. The authors analyze the influence of every step in web engineering process on a final product, and suggest that evaluation process should be performed after each step, with respect to quality models defined for each of them.

Since ISO 9126 is a widely adopted and used standard, we have also adopted it for constructing the quality model for semantic technologies. For building such quality model, we have used a bottom-up approach.

The following sections describe in detail how we have extracted the quality model for semantic technologies from existing evaluations.

4 Defining Quality Measures

The starting point for defining software quality measures has been the set of evaluation results obtained in the SEALS European project[1] , which provides evaluation results for different types of semantic technologies (ontology engineering tools [20], reasoning systems [21], ontology matching tools [22], semantic search tools [23], and semantic web service tools [24]).

Based on the evaluation results for a particular software product and the analysis of these results, a set of quality measures was defined. Then, some primitive measures were combined to obtain derived ones. For any derived quality measure defined,we specified the function (or set of functions) that allows obtaining such derived measure from the primitive ones.

Different evaluation scenarios were defined for each type of technology and in each of them different types of test data were used as input. Evaluation raw results represent the data obtained as the output of the evaluation when using some test data, and they are considered to be primitive measures. These test data and raw results provide us with enough information for defining the hierarchy of software quality measures.

For instance, for evaluating the conformance of ontology engineering tools, the experiment consisted in importing the file containing an ontology (O_i) into the tool and then exporting the imported ontology to another file (O_i^{II}) (Fig. 2 shows the steps of the conformance evaluation).

In this evaluation test suites with ontologies modeled in different ontology languages were used. Each test in a test suite contains:

– *Origin ontology.* The ontology to be used as input.

The raw results of one test execution are:

– *Final ontology.* The ontology that is produced by the tool when importing and exporting the origin ontology.
– *Execution Problem.* Whether there were any execution problems in the tool when importing and exporting the origin ontology. Possible values are *true*, and *false*.

Based on the test data and the raw results of one test execution, the following interpretations and their functions were made:

[1] http://www.seals-project.eu

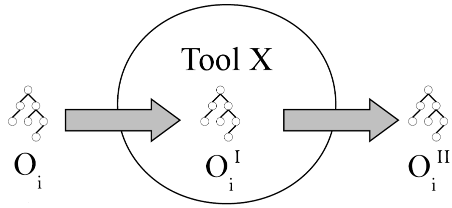


Fig. 2. Steps of a conformance test execution

- *Information added.* The information added to the origin ontology after importing and exporting it.

$$\text{final ontology} \ - \text{origin ontology}$$

- *Information lost.* The information lost from the origin ontology after importing and exporting it.

$$\text{origin ontology} \ - \text{final ontology}$$

- *Structurally equivalent.* Whether the origin ontology and the final one are structurally equivalent. Possible values are *true*, and *false*.

$$(\text{information added} = \text{null}) \wedge (\text{information lost} = \text{null})$$

- *Semantically equivalent.* Whether the origin ontology and the final one are semantically equivalent. Possible values are *true*, and *false*.

$$\text{final ontology} \equiv \text{origin ontology}$$

- *Conformance.* Whether the ontology has been imported and exported correctly with no addition or loss of information. Possible values are *true*, and *false*.

$$\text{semantically equivalent} \wedge \neg(\text{execution problem})$$

Those interpretations are actually derived measures obtained from one particular test, and are further combined in order to obtain measures for the whole test suite. From the derived measures in the conformance scenario, the following test suite measures were obtained:

- *Ontology language component support.* Whether the tool fully supports an ontology language component.

$$\frac{\#\text{ tests that contain the component where conformance} = \text{true}}{\#\text{ tests that contain the component}} = 1$$

- *Ontology language component coverage.* The ratio of ontology components that are shared by a tool internal model and an ontology language model.

$$\frac{\#\text{ components in the ontology language where component support} = \text{true}}{\#\text{ components in the ontology language}} \times 100$$

– *Ontology information persistence.* The ratio of information additions or loss when importing and exporting ontologies.

$$\frac{\#\text{ tests where information added} \neq \text{null or information lost} \neq \text{null}}{\#\text{ tests}} \times 100$$

– *Execution errors.* The ratio of tool execution errors when importing and exporting ontologies.

$$\frac{\#\text{ tests where execution problem} = \text{true}}{\#\text{ tests}} \times 100$$

Similarly to the example of the conformance evaluation for ontology engineering tools presented above, we have defined measures for the other types of tools. Table 3 summarizes the obtained results.

Table 3. Total number of measures obtained for semantic technologies

Tool/Measures	**Raw results**	**Interpretations**	**Measures**
Ontology engineering tools	7	20	8
Ontology matching tools	1	3	4
Reasoning systems	7	0	7
Semantic search tools	13	16	18
Semantic web service tools	2	3	3

5 Defining Software Quality Sub-characteristics

Every software product from a particular domain has some sub-characteristics that are different from other software systems and those sub-characteristics, together with more generic ones, should be identified and precisely defined. Every quality measure provides some information about one or several software sub-characteristics; therefore, based on the software quality measures that we defined, we specified a set of software quality sub-characteristics. In some cases, a quality sub-characteristic does not have only one measure that determines it, but a set of measures. Finally, some quality sub-characteristics were combined into more general ones.

In the example of the conformance scenario for ontology engineering tools, based on the measures and analysis presented in previous section, we have identified two quality sub-characteristics that are specific to this type of software, which are the following:

– *Ontology language model conformance.* The degree to which the knowledge representation model of the software product adheres to the knowledge representation model of an ontology language. It can be measured using two different measures, which are obtained in the conformance evaluation:

  - *Ontology language component coverage*
  - *Ontology language component support*
- *Ontology I/E accuracy.* The accuracy of the process of importing and exporting ontologies. It can be measured using:
  - *Ontology information persistence*

Furthermore, we have identified one quality sub-characteristic which is general and can be used for different kinds of software:

- *Robustness.* The ability of the software product to function correctly in the presence of invalid inputs or stressful environmental conditions. It can be measured using:
  - *Execution errors*

Figure 3 presents the raw results, interpretations, quality measures and quality characteristics of the conformance evaluation for ontology engineering tools.

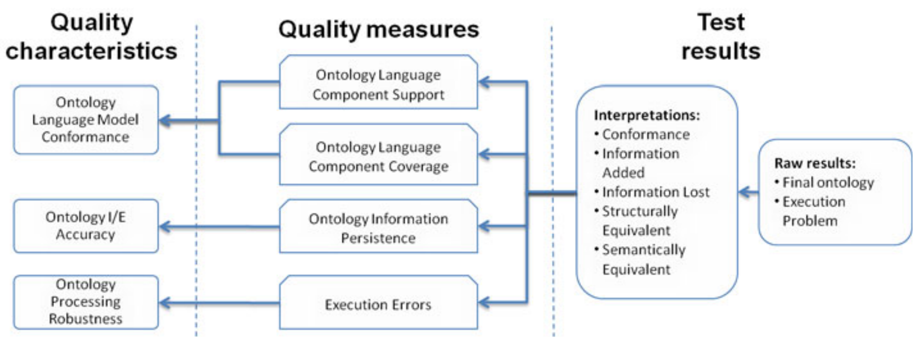


Fig. 3. Entities in the conformance scenario for ontology engineering tools

In total, we have identified twelve semantic quality sub-characteristics:

- *Ontology Language Model Conformance.* The degree to which the knowledge representation model of the software product adheres to the knowledge representation model of an ontology language.
- *Ontology Language Interoperability.* The degree to which the software product can interchange ontologies and use the ontologies that have been interchanged.
- *Reasoning Accuracy.* The accuracy of the reasoning process.
- *Ontology Alignment Accuracy.* The accuracy of the matching process.
- *Semantic Search Accuracy.* The accuracy of the semantic search process.
- *Semantic Web Service Discovery Accuracy.* The accuracy of the process of finding services that can be used to fulfil a given requirement from the service requester.
- *Ontology I/E Accuracy.* The accuracy of the process of importing and exporting ontologies by the tool.

- *Ontology Interchange Accuracy.* The accuracy of the interchange of ontologies between tools.
- *Query Language Suitability.* The ability of the tool to provide appropriate queries for a given questions.
- *Ontology Processing Time Behaviour.* The capability of the software product to provide appropriate response and processing times when working with ontologies.
- *Reasoning Time Behaviour.* The capability of the software product to provide appropriate response and processing times when performing reasoning tasks.
- *Semantic Search Time Behaviour.* The capability of the software product to provide appropriate response and processing times when performing the search task.

6 Aligning Quality Sub-characteristics to a Quality Model

In the previous step we have identified a set of quality sub-characteristics that are specific for semantic technologies. After that, the alignment with the ISO 9126 quality model was established; i.e., all the identified sub-characteristics were properly assigned to those already specified in the ISO 9126 quality model.

For instance, *Ontology language model conformance* is defined as a sub-characteristic of *Functionality compliance* (the capability of the software

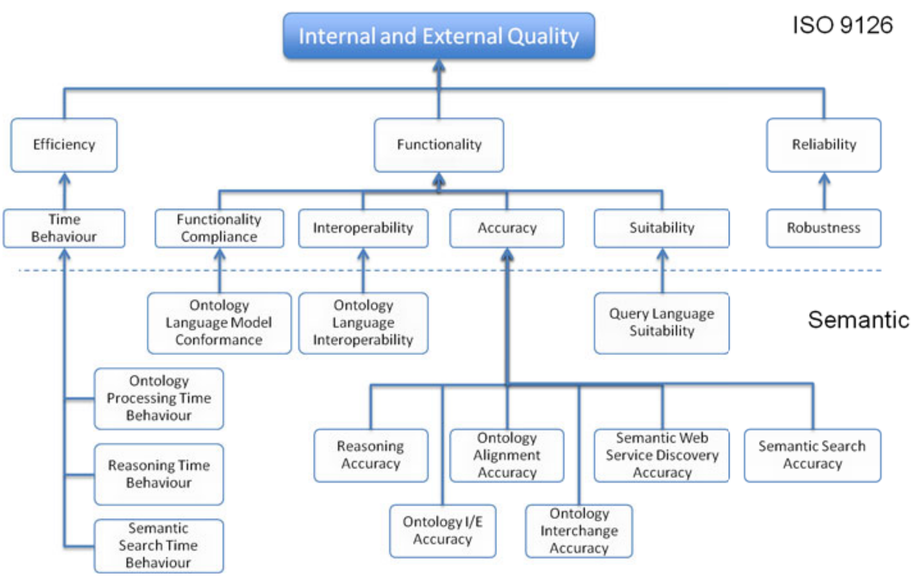


Fig. 4. External and internal quality characteristics for semantic technologies

product to adhere to standards, conventions or regulations in laws and similar prescriptions relating to functionality).

Fig. 4 shows the proposed quality model for semantic technologies, while Fig. 5 shows quality in use.

Fig. 5. Quality in use quality characteristics for semantic technologies

7 Conclusions and Future Work

This paper presents a first step towards a quality model for semantic technologies, which extends the ISO 9126 software quality model. Such quality model can provide a framework for the evaluation and comparison of semantic technologies.

For building the quality model, we have used a bottom-up approach. It starts from the evaluations that have already been performed and continues towards defining quality measures. When a set of quality measures is defined, quality sub-characteristics are specified together with their hierarchy. At the end, sub-characteristics are aligned to an existing quality model.

The model presented in this paper is not complete. It only includes quality entities that are based on some existing evaluations. In the future, the model will be completed either by extending the evaluations, or by applying a top-down approach and specifying additional quality entities that are currently not defined in the model.

Some authors include in their software quality models relationships between quality entities [11,14]. The inclusion of these entities into our model will be addressed in a future iteration.

Furthermore, the ISO 9126 standard is to be replaced by the SQuaRE one, and when the SQuaRE software quality model becomes available, the proposed quality model for semantic technologies should be adapted to it.

One future use of the quality model presented in this paper, based on the evaluation results that are being obtained in the SEALS project, is to build a recommendation system for semantic technologies that will allow extracting semantic technology roadmaps. This will provide users with guidance and recommendation of the semantic technologies that better suite their needs.

Acknowledgments

This work is supported by the SEALS European project (FP7-238975) and by the EspOnt project (CCG10-UPM/TIC-5794) co-funded by the Universidad Politécnica de Madrid and the Comunidad de Madrid.

References

1. Behkamal, B., Kahani, M., Akbari, M.K.: Customizing ISO 9126 quality model for evaluation of B2B applications. Information and Software Technology 51(3), 599–609 (2009)
2. OntoWeb: Ontoweb deliverable 1.3: A survey on ontology tools. Technical report, IST OntoWeb Thematic Network (2002)
3. Guo, Y., Pan, Z., Heflin, J.: LUBM: A benchmark for OWL knowledge base systems. Web Semantics: Science, Services and Agents on the World Wide Web 3(2-3), 158–182 (2005)
4. Lambrix, P., Habbouche, M., Perez, M.: Evaluation of ontology development tools for bioinformatics. Bioinformatics 19, 1564 (2003)
5. García-Castro, R., Gómez-Pérez, A.: Interoperability results for Semantic Web technologies using OWL as the interchange language. Web Semantics: Science, Services and Agents on the World Wide Web 8, 278–291 (2010)
6. Garvin, D.A.: What does" product quality" really mean? Sloan Management Review 26(1), 25–43 (1984)
7. Kitchenham, B., Pfleeger, S.L.: Software quality: the elusive target [special issues section]. IEEE Software 13(1), 12–21 (1996)
8. Cavano, J.P., McCall, J.A.: A framework for the measurement of software quality. ACM SIGMETRICS Performance Evaluation Review 7(3-4), 133–139 (1978)
9. Boehm, B., Brown, J., Lipow, M.: Quantitative evaluation of software quality. In: Proceedings of the 2nd International Conference on Software Engineering, pp. 592–605. IEEE Computer Society Press, Los Alamitos (1976)
10. ISO: ISO/IEC 9126-1:2001, Software engineering – Product quality – Part 1: Quality model. Technical report, International Organization for Standardization (2001)
11. Botella, P., Burgués, X., Carvallo, J., Franch, X., Pastor, J., Quer, C.: Towards a quality model for the selection of ERP systems. Component-Based Software Quality, 225–245 (2003)
12. Azuma, M.: SQuaRE: the next generation of the ISO/IEC 9126 and 14598 international standards series on software product quality. In: ESCOM (European Software Control and Metrics Conference), pp. 337–346 (2001)
13. Dromey, R.G.: Software Product Quality: Theory, Model, and Practice. Software Quality Institute, Brisbane, Australia (1998)
14. Franch, X., Carvallo, J.P.: Using quality models in software package selection. IEEE Software 20(1), 34–41 (2003)
15. Carvallo, J., Franch, X., Quer, C.: Defining a quality model for mail servers. In: Erdogmus, H., Weng, T. (eds.) ICCBSS 2003. LNCS, vol. 2580, pp. 51–61. Springer, Heidelberg (2003)
16. Zulzalil, H., Ghani, A.A.A., Selamat, M.H., Mahmod, R.: A Case Study to Identify Quality Attributes Relationships for Web-based Applications. IJCSNS 8(11), 215 (2008)
17. Padayachee, I., Kotze, P., van Der Merwe, A.: ISO 9126 external systems quality characteristics, sub-characteristics and domain specific criteria for evaluating e-Learning systems. In: The Southern African Computer Lecturers' Association, University of Pretoria, South Africa (2010)
18. Dominguez-Mayo, F., Escalona, M., Mejias, M., Torres, A.: A Quality Model in a Quality Evaluation Framework for MDWE Methodologies. In: 2010 Fourth International Conference on Research Challenges in Information Science (RCIS), pp. 495–506. IEEE, Los Alamitos (2010)

19. Cachero, C., Poels, G., Calero, C.: Towards a quality-aware Web Engineering process. In: Proceedings of the 12th International Workshop on Exploring Modeling Methods in Systems Analysis and Design (EMMSAD 2007), Citeseer, pp. 7–16 (2007)
20. García-Castro, R., Grimm, S., Toma, I., Schneider, M., Marte, A., Tymaniuk, S.: D10.3 Results of the first evaluation of ontology engineering tools. Technical report, SEALS Consortium (2010)
21. Yatskevich, M., Marte, A.: D11.3 Results of the first evaluation of advanced reasoning systems. Technical report, SEALS Consortium (2010)
22. Euzenat, J., Meilicke, C., Trojahn, C., Šváb Zamazal, O.: D12.3 Results of the first evaluation of matching tools. Technical report, SEALS Consortium (2010)
23. Wrigley, S.N., Elbedweihy, K., Reinhard, D., Bernstein, A., Ciravegna, F.: D13.3 Results of the first evaluation of semantic search tools. Technical report, SEALS Consortium (2010)
24. Tymaniuk, S., Cabral, L., Winkler, D., Toma, I.: D14.3 Results of the first evaluation of Semantic Web Service tools. Technical report, SEALS Consortium (2010)

Influence of Human Factors in Software Quality and Productivity

Luis Fernández-Sanz and Sanjay Misra

[1] Universidad de Alcalá, Depto. de Ciencias de la Computación,
Alcalá de Henares, 28400 Madrid
[2] Atilim University, Ankara, Turkey
luis.fernandezs@uah.es, ssopam@gmail.com

Abstract. Traditionally more effort has been devoted to technical and process aspects of software quality and productivity. However, in an activity like software development so intensive in workforce and so dependent on professionals' performance, it is strange that human and social factors affecting development teams have been attracted little attention. This paper is aimed at analyzing contributions in this area as well as providing empirical data from specific initiatives to know more about real practices and situation in software organizations.

Keywords: software quality, human factors, software productivity.

1 Introduction

From early times of software engineering, attention to software quality has been evident. The so-called "software crisis" was defined as a critical situation where low productivity, poor quality, budget slippage and imprecise planning and cost estimation provoked the birth of the concept of software engineering [1][2], joining this way quality and productivity. Different studies usually highlight the quality problems which arise in software development by collecting data from projects and systems (e.g. the Chaos reports [3] or the Risk to the Public reports on incidents in software and systems (http://www.csl.sri.com/users/neumann/neumann.html); even there are specific European studies like [4] or [5].

Certainly, software development activity differs clearly from manufacturing of other types of products: besides intrinsic differences between software and other products [1], processes are different from the most traditional manufacturing of products [6]. As an example, software development is one of the most specific activities which require its own adaptation guide (ISO 9001-3) [7] for implementing a so general standard for quality like ISO 9001 [8]. This is one of the reasons why it has not been possible to solve the need of techniques and methods for software quality assurance by a direct and quick translation of mature quality practices in other sectors. This has leaded to a huge effort for getting a trustable adaptation of activities for other products' development as well as for generating new and specific models, processes and methods for software development.

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 257–269, 2011.

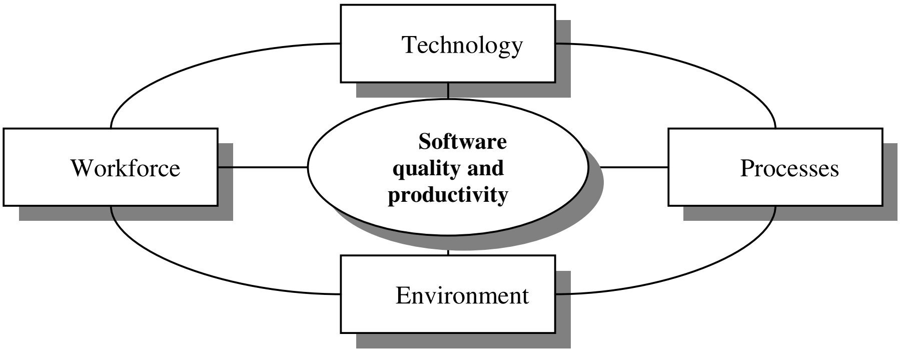


Fig. 1. Factors for software quality and productivity [9]

If we want to analyze which are the sources of quality and productivity, the model presented in [9] (see Fig. 1) would help us to allocate current lines of actions in big categories according to their core philosophy.

First one is technology improvement and evolution. This is an evident option for almost all professionals, sometimes boosted by the commercial activity behind each new technical option launched to market. Operating systems, platforms, languages, etc. in a general view but also improved functionality for developers' tools (IDE, CASE, etc.) and new architectures and paradigms. Deplorably, it is usually hard to find out independent and rigorous studies with data from real practice which support performance improvements proclaimed by vendors (maybe because they have not been measured or due to hidden difficulties). In general, better tools tend to increase productivity: it is supposed that advanced tools could lead to –17% of effort while extremely basic tools would cause a +24% [10]. In any case, true potential for this line of action is heavily linked to real use by practitioners which should be promoted by implementing a formal and customized process of acquisition, implementation and training. Sadly, organizations tend to be more conscious in purchasing in tools (more than 50% according to [9]) than in investing in other actions which might be more efficient. Effective CASE approach increases productivity in 27% but an inadequate implantation would lead to a fall of 75% and it is remarked the danger of exaggerated and sensational ads, that 10% of them are purchased but never used, that 25% are poorly exploited due to lack of training, etc. [7]. Even in [11] it is shown that 75% of statements in ads are considered as ones with low credibility although, after reviewing more than 4000 projects, 70% of people in charge of projects believe that one unique factor like this would provide big improvements.

The second category is processes. This a huge category as it embraces both processes in the large (with a special focus on quality assurance and management activities) and specific methods and techniques for detailed tasks at project level (also with specific interest on quality-related activities). In the first group, the main effort has been devoted to the well-known process models like CMMi [12] and ISO15504 [13]. Lastly agile methods (e.g. SCRUM [14]) have been emerged with energy as proposed solutions for quality and productivity while more traditional proposals like Unified Process [15] or classical methodologies and life cycles models are not today choices if not embedded within bigger frameworks, specially, because as opposed to them, they have been well accepted but almost no real rigorous data on benefits have been provided (at least, different studies on CMMi influence have been published during the last years, e.g. [16][17]). Within the proposals of processes in the large, the specific proposals centred on quality assurance are also important chances, specially the adaptation of the general ISO 9001 [8] to software development [7]. Although its implementation in the software companies has been scarce and with a total absence of measurement of benefits, it is true that its consequences in organizations (systematic documented processes, etc.) represent a minimum application of quality practices which provide some trust in reaching project objectives.

In the category of processes at project level, specific quality plans (e.g. IEEE [18]) usually rely on a traditional set of techniques to help in quality assurance: configuration management, metrics/measurement, verification and validation (highlighting software testing and review and audits processes as commonest methods although additional techniques are available but not very common [19]). It is important to remark that these techniques require non trivial investment in human effort (hence money) so their use should be adapted to each organization both to be effective (e.g. providing an easy-to-use environment) and to be efficient and cost-effective (e.g. reviewing all the code is usually unfeasible so a previous selection of defect-prone parts using metrics and Pareto principle and the help of automatic reviewing tools are a must). In general, adequate quality assurance efforts tend to make more stable delivery schedules and quality levels [17]. A number of studies (e.g. [20] [21]) show productivity improvement and good ROI based on decreasing rework and bad quality costs [22], especially when focus is put on prevention and early detection because fixing cost rises sharply when defects remain undetected throughout the project phases [1][23].

Finally we have environment and personnel as the two factors which are by far the least analyzed in a formal way, specially their relation to effects on software quality and productivity described in a measurable manner [24]. In fact, these two issues (environment and personnel) are the key factors in agility which is a very common practice in software development, increasingly adopted by the software community. The people factors play an important role in agile software development. In fact, the individual competency is a critical factor for the success of agile projects where people is working close to each other to better interact and solve the problems through team competence too. This is the reason agile processes are designed to capitalize on each individual and each team's unique strength. Moreover, Cockburn and Highsmith [25] have emphasized on human behaviours and stated that the most important implication to managers working is that it places more emphasis on people factors in the project. They further classified the people factors in amicability, talent, skill, and communications which are the primary concerns in agile development. Furthermore, the people, environment, and organizational culture all influence one another and have an impact on quality of product and therefore make the basis for the success of project.

Moreover a quote in software engineering states that performance differences have been proved among software professionals even in the conditions of identical task [26]. Although the software community is aware of this fact, i.e. the talent has great effect on their success, still most of the software development organizations are focusing so much on tools and technology and little on people. Even more Pressman

[1] also supports the above statement by giving an quote of a Vice president of a big company who stated that the most important ingredient that was successful in on this project was having smart people: "The most important ingredient that was successful on this project was having smart people…very little else matters in my opinion….The most important thing you do for a project is selecting the staff…The success of the software development organization is very, very much associated with the ability to recruit good people".

All these studies reflect that impacts of human factors are dramatic in the success of software development but unfortunately they do not attract proper attention. This motivates us to work in this area for investigation focused on the study of the influence of human factors in software quality and productivity.

In the following sections of this paper existing studies on influence of personnel characteristics and environment for professionals in software quality and productivity will be analyzed. In section 3, results from authors' studies on some specific points referred to software development personnel are presented. Discussion, conclusions and future lines of actions are presented in sections 4 and 5.

2 Human Factors

As development cost models have shown, humans represent the main resource (and the main cost) for software projects. Although specific studies on software workforce are not usual in literature, there are several references of interest to at least devise an overview of lines of action.

In general, proposals related to cost estimation and project management have been the pioneers in the analysis of human and environmental factors due to their influence on budget, schedule and quality. One sophisticated example of modelling of relations among the different factors which influence software development and effort and results during projects is the project dynamics proposed by Abdel-Hamid [27]. Based on this model, subsequent efforts have checked if the famous Brook's law [28], "adding people to a delayed project tends to delay it more and with higher cost", is confirmed with results of the model: one study [29] confirms higher costs but not always more delay (only when tasks are highly sequential), others restrict the effects to small projects below 5 people involved. Of course, this can be pointed out using the traditional study of Schulmeyer [30] where the phenomenon of the net negative producing programmer is exposed. In a recent work [31] the authors have shown that size of team and experience/expertise of the project team were found the predictive effect in thinking style (simplicity) of project management in terms of being prescient, proactive, retrained and productive.

However, locating quantified analysis of development personnel is usually linked to contribution in the area of software cost estimation models. An evident reference in this approach is the quantification of influence of qualification and experience of development personnel in the traditional COCOMO drivers [20]: analysts' capacity, experience with similar applications, programmers' capacity, experience with virtual machine and experience with language. Letting aside the precision and validity of COCOMO, it is interesting to realize that negative influence of these factors is always greater than the corresponding positive influence when having better than average situations (e.g. low valued analysts represent 46% of extra cost while highly skills analysts contribute with a 29% reduction of costs).

Another interesting set of data on the influence of factors related to software development personnel was published by C.Jones [32] within his wide statistics on software costs, mainly based on function points. Table 1 extracted from [32] shows the different influence of key factors when situation is positive (e.g., very experienced personnel: +55%) and when it is negative (e.g., -87%). In general, accumulation of negative factors in environment and personnel means a big fall of productivity in much higher degree than positive factors. As a consequence, the sentence "people are our main asset" is even more justified in software development, at least avoiding not cutting costs too much in this aspect.

Table 1. Impact of key factors affecting productivity

Factor	Positive influence (+%)	Negative influence (-%)
Developers' experience	High +55%	Low –87%
Lack of experience managers	High +65%	Low –90%
Office	Ergonomic +15%	Crowded –27%
Not paid extra hours	Yes +15%	No 0%
Moral	High +7%	Low –6%
Organization	Hierarchical +5%	Matrix –8%
Annual training	>10 days +8%	No training -12%
Schedule pressure	Moderate +11%	Excessive -30%

As seen moral and motivation is a key factor. Research into motivation of software developers suggests they might resist the implementation of quality management techniques [33][34][35]. In general, there are many sources of motivation and demotivation for software engineers [36] and motivation is recognized as a major influence for quality [24] with many studies analyzing level of motivation of software developers (compared to other professions or positions) and their resistance to change (really high in developers engaged in software maintenance [37]) but with a favourable influence of human relations on motivation [38]. In fact, the use of disciplined teams for development (with disciplined methods) tends to produce better results than ad-hoc or individual programmers [39]. This is also aligned to models more oriented to personal performance of developers like PSP [40]. Given that one important demotivator is producing poor quality software with no sense of accomplishment [41], total freedom or ad-hoc arrangements would be not a real motivation for software professionals.

When dealing with software quality and reliability, human factors have been identified as a cause of problems which can determine success or not of a project or system [38]. As stated in several articles [42] [43], the quality of the people should be considered the primary driver for software quality while sometimes too much industry focus has been on the process ("[software] It's more about people working together than it is about defined processes"). From an intuitive and experiential perspective, the education and abilities of a developer represents an important part in the ultimate quality of their developed software but little empirical evidence to support this logical assumption. One of the scarce studies analyzing this relationship has shown that higher proportions of skilled engineers had the most dramatic effect in terms of adequacy of the design and implementation while higher proportions of less skilled engineers negatively affected the end product quality of the software [44]. However, the study allocates the most dramatic positive effect to functional completeness to experienced leader-ship in all stages of the project. Some additional experiments [45][46][47][48] tend to show a correlation between good teamwork dynamics or other indicators (e.g. size between 5 and 7) and success and quality in the corresponding projects.

It is clear that attitude is an important factor for both quality and productivity that might be also linked to professional ethics codes [49]. Specially attitude of testers and developers towards testing is considered critical for software quality [50][51][52].

Obviously one key point is the adequacy of education, qualification and soft skills for the corresponding position. It is difficult to have clear definitions for requirements and skills needed for each role or position. Although big efforts on collecting information from, e.g., job ads enable certain general descriptions [53], it is hard to quantify qualification and experience of people in order to find out possible relationships to effects in productivity and quality with enough accuracy. However, as stated in data included in Table 1, training is one of the key factors which can be evaluated by collecting data from professionals as well as the evaluation of developers in terms of factors which influence and their perception of their own skills. In the next section, some results from surveys to professionals will be presented.

The development of people capability maturity model (People CMM) is an attempt to improve the performance and level of organisations through of people of organisation. Its first version [54] was released in 1995 and after the continuous assessment (applicability to the industry), the modified version was released in 2002 [55]. People CMM has also five stages (like CMMI) with successive foundations for continuously improving workforce competencies. This process start in stage 1 where workforce practices are performed inconsistently or ritualistically and frequently fail to achieve their intended purpose; and it is oriented to reach the Optimizing Maturity Level (Level 5)[56]. In fact accreditation from people CMMI is a way to improve the workforce performance in a systematic way.

3 Results from Specific Surveys

As a contribution to this field of research, the results of several surveys involving representative samples of software professionals are presented in this section. The surveys are oriented to understand which is the situation of real testing practice and which factors related to professionals (attitude, training or similar items) are having a real influence in software quality in terms of the perception of participants. As starting point for this analysis, we should address some previous results from a survey aimed at knowing which factors have a negative influence on real practice of software testing. Preliminary results were published in [57]. The published results involved a varied sample of 127 practitioners to collect opinion on the proposed list of factors of influence (see Table 2).

Table 2. Sample data for survey on factors with negative influence on testing

Sector	%	Position	%
Government	22.9%	Project manager	21.8%
Telco and IT	21.8%	Tester	14.6%
Consultancy	14.5%	Manager	14.6%
Finance	9.8%	Programmer	12.5%
Defense	5.2%	QASpecialist	9.4%
Tourism	5.2%	Systems analyst	7.3%
Health	3.1%	Others	20.3%
Transport/Airlines	3.1%		
Others	14.4%		

The survey included a set of 23 factors determined by a panel of experts from academia and industry. Respondents had to indicate if they consider each item an effective factor of influence; they also ranked influence in a three-level scale: total/partial/ none. Although detailed results are available in [57], for the purpose of our analysis in this paper, we want to highlight the following ones:

- Many software professionals have not received any type of specific training in software testing: 93,7% of participants confirmed this assertion and only 20,47% said this does not have any negative influence on real practice.
- Many managers did not attend good training on software testing so they do not appreciate its interest or potential for efficiency and quality: 85,8% of participants confirmed this assertion and only 21,26% said this does not have any negative influence on real practice.
- People tend to execute testing in an uncontrolled manner until the total expenditure of resources in the belief that if we test a lot, in the end, we will cover or control all the system: 58,3% of participants confirmed this assertion and only 37,01% said this does not have any negative influence on real practice.
- It is not usual to plan and design efficient cases with minimum cost or to link tests to priorities or risks; there is not control on incurred risks depending on tests, no control of evidences, etc.: 74,0% of participants confirmed this assertion and only 25,98% said this does not have any negative influence on real practice.

Although these results are valuable, it is important to contrast them with equivalent alternative sources of information. In the case of training, the following data have been collected:

- An informal online surveys by http://www.methodsandtools.com in 2005 with 240 respondents confirmed low percentage of testing training in their lifetime: none 43% and less than a week 18%. Another survey in 2009 in the same website resulted in a 23% of the 258 respondents having not received any training in the last three years and 19% less than a week.
- An specific online survey promoted by the Software Quality group of ATI (www.ati.es/softwarequalitygroup), the main Spanish II professionals association, involved 42 professionals specifically invited to answer in 2009.

50% of them did not receive any type of training (including informal in-company training but excluding self-study) in their life time.

- A direct survey with questionnaires directly conducted by L.Fernández-Sanz during specific events (courses, conferences, etc.) included 210 software professionals (see sample data in Figure 1). This reported that only 42.68% of 147 respondents with an average experience in their position of 4.8 years (the question on training was not included in first questionnaires) received specific training in software testing. Moreover, analyzing training per each group of respondents, 44% of managers declared specific training, 66% of project leaders, 100% of quality assurance (QA) managers while only 32% of developers and 68,4% of testers.

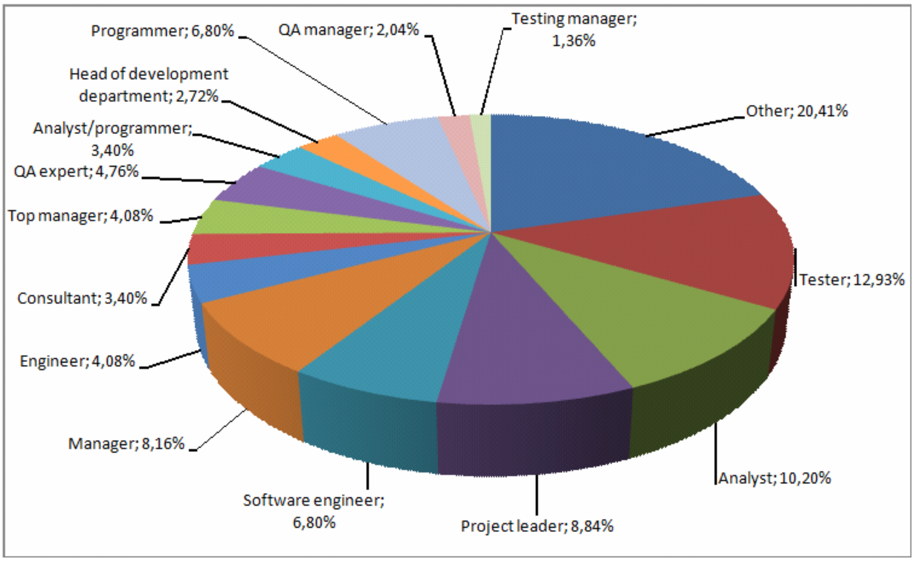


Fig. 2. Sample distribution for training survey

- A preliminary study [58] to evaluate how software professionals design test cases when facing a simple case study show that a sample of 71 participants tended to carry out design in an unsystematic way without a control of functional coverage (only one covered 75% of options and 56% of participants covered less than 50%) and repeating unnecessarily a high percentage of cases (around 50% of proposed test cases). An additional result comes from the fact that testers were required to prioritize the different options or paths in the system interaction according to their view of importance. It was observed that among the 10 most executed test cases, there was only one of the ten most important ones and there were 3 of the most important ones among the 10 least executed cases. So it is clear that test case design was done without prioritizing cases according to importance or risk for system.

These data seem to confirm that opinion of professionals expressed in a survey on possible factors with negative influence on testing practice are consistent with data independently collected from other samples of software people. Low percentage of people with specific training confirms some of the conclusions of the survey on factors of influence and data from experiment on test case design confirms the other ones. However, perceptions expressed in the factors study are not always so clearly confirmed: although the sample is not wide and it is collected in environments where there is some underlying interest in software quality (events and courses in software engineering and quality), percentage of managers or project leaders with declared specific training in testing is not low although samples of each group are not wide (e.g. 25 managers, 5 QA managers, etc.). Another idea to be taken into account is a possible bias in managers' answers although questionnaires were anonymous: it could be embarrassing even in private environment to accept that you do not have this specific training being a manager.

In the case of practices related to test design, data from experiment are clear: participants tended to cover functionality by accumulating cases without systematic philosophy without minimum cost (repeating equivalent cases several times with low coverage) and without priority related to risks or importance.

This cross-checking is really valuable to contrast data: what at the beginning could only be the opinion of experts suggesting causes for bad testing practices and what a good sample of professionals say about factors suggested by experts could be only a opinion but when another independent sample of professionals confirms general trends, it should be considered more than a opinion of people.

4 Discussion

Software testing is the first software quality assurance technique which controls the quality of software product [59]. Although there are other techniques which is supposed to implement from the beginning of the development e.g software review, walkthrough, inspection etc., these practices are not very common and acceptable especially from medium scale companies due to several reasons [60]. As a consequence, testing is the only way to control the quality of product. In other words, testing is undoubtedly the largest consumer of software quality. Galin [59] has reported a survey report which states that 24% of project development budget, 32% of the project management budget is used for allocation of testing process and, even more, the 45% of project time was scheduled for testing. These observations explain the importance of this most important quality assurance activity.

However, observations in the previous sections reflect that this activity is not taken very seriously by the management of software companies despite its importance and we further observed that majority of the testers, i.e. not only specialists but mainly developers and other personnel involved in the project who has to contribute to some of the testing levels, have not received proper training. We have also observed how the human factors seriously affect on the performance of the team members; in this case, if the testing team is not being benefited from proper training in this area, they won't be able to perform well.

5 Conclusions

In this article factors affecting software quality and productivity have been discussed from the point of view of existing contributions which have tried to evaluate and quantify their possible influence on software. In the traditional view of three Ps (Product, process and people) or in the one excerpted from [9] with process, technology, people and environment, we have revisited contributions focusing our attention on the human side, the role played by software professionals, the environment where they work, their qualification and skills, the motivation and attitude, etc. Although studies in this line of action are far from being as frequent as the ones centred on techniques, process models and tools, a good number of research works have been published highlighting the clear influence of the human-related factors.

In order to contribute to this field, we have presented results from different surveys which collected information from a good sample of software professionals in order to know more on human factors affecting real practice in software testing. Starting from preliminary results on factors with a negative influence on testing practice, we have collected additional sources of data which have confirmed some of the conclusions about training in software testing as well as non systematic test design practices. We are planning to collect more data from additional sources and publications as well as from specific activities with professionals in order to devise a complete and precise panorama of real testing practices and how human factors involving professionals engaged in software projects might influence quality and productivity.

References

1. Pressman, R.S.: Software Engineering Software Engineering: A Practitioner's Approach. McGraw-Hill, New York (2004)
2. Fenton, N.E.: Software metrics: a rigorous and practical approach. Thomson, Boston (1997)
3. Standish Group: Chaos Chronicles v 3.0. Standish Group, West Yarmouth (2004)
4. Möller, K.H., Paulish, D.J.: Software metrics: A practitioner's guide to improved product development. Chapman & Hall, Londres (1993)
5. Stockman, S.G., Norris, M.T.: Eng. Approaches to Soft. Quality Engineering. In: Fenton, N., Littlewood, B. (eds.) Soft. Reliability and Metrics, Elsevier Science, Londres (1991)
6. Chistenson, D.A.: Statistical methods applied to software. In: Schulmeyer, G.G., McManus, J.I. (eds.) Total Quality Management for Software. Van Nostrand, New York (1992)
7. ISO, ISO/IEC 90003:2004, Software engineering - Guidelines for the application of ISO 9001:2000 to computer software, ISO, Geneve (2004)
8. ISO, ISO 9001:2008, Quality management systems - Requirements, ISO, Geneve (2008)
9. McConnell, S.: Rapid Development: Taming Wild Software Schedules. MS press, Redmond (1996)
10. Boehm, B.W.: Software engineering economics. Prentice-Hall, New Jersey (1981)
11. Jones, C.: Assessment and control of software risks. Yourdon Press, Saddle River (1994)
12. CMMI Product Team, CMMI for Development, Version 1.2,CMMI-DEV, V1.2, CMU/SEI-2006-TR-008, ESC-TR-2006-008. Software Engineering Institute, Pittsburgh (2006)

13. ISO, ISO 15504-1:2004, Information technology - Process assessment – Part 1: Concepts and vocabulary, ISO, Geneve (2004)
14. Cohn, M.: Succeeding with Agile: Software Development Using Scrum. Addison-Wesley, Reading (2009)
15. Jacobson, I., Booch, G., Rumbaugh, J.: The Unified Software Development Process. Addison-Wesley, Reading (1999)
16. Goldenson, D.R., Gibson, D.L.: Demonstrating the Impact and Benefits of CMMI: An Update and Preliminary Results, CMU/SEI-2003-SR-009. Software Engineering Institute, Pittsburgh (2003)
17. Herbsleb, J., et al.: Benefits of CMM-based software process improvement: initial results. CMU-SEI-94-TR-013. Software Engineering Institute, Pittsburgh (1994)
18. IEEE: Std. 730, Standard for Software Quality Assurance Plans. IEEE, New York (1984)
19. Adrion, W.R., Branstad, M.A., Cherniavsky, J.C.: Verification, Validation, and testing of Computer Software. ACM Computing Surveys 14(2), 159–192 (1982)
20. van Solingen, R., Rico, D.F.: Calculating Software Process Improvement's Return on Investment. Advances in Computers 66, 2–43 (2006)
21. Rico, D.F.: ROI of Software Process Improvement: Metrics for Project Managers and Software Engineers. J. Ross Publishing (2004)
22. Fleckenstein, W.O.: Challenges in software development. Computer (3), 60–64 (1983)
23. Harrington, H.J.: Poor–quality cost. McGraw-Hill, New York (1987)
24. Thomas, S.A., Hurley, S.F.: Looking for the human factors in software quality management. In: Proceedings of the 1996 International Conference on Software Engineering: Education and Practice (SEEP 1996), pp. 474–480. IEEE Computer Society, Washington, DC, USA (1996)
25. Cockburn, A., Highsmith, J.: Agile software development: A people factor. Software Management 34, 131–134 (2001)
26. Mishra, A., Misra, S.: People management in software industry: the key to success. SIGSOFT Software Engineering Notes 35, 1–4 (2010)
27. Abdel-Hamid, T.K., Madnick, S.: Software project dynamics. An integrated approach. Prentice-Hall, New Jersey (1991)
28. Brooks, F.: The mythical man-month. Addison-Wesley, Boston (1975)
29. Hsia, P., Hsu, C.-T., Kung, D.C.: Brooks' Law Revisited: A System Dynamics Approach. In: 23rd International Computer Software and Applications Conference (COMPSAC 1999), pp. 370–375. IEEE Computer Society, Washington, DC, USA (1999)
30. Schulmeyer, G.G.: The net negative producing programmer. American Programmer (6) (1992)
31. Akman, I., Misra, S., Cafer, F.: The role of Leadership cognitive complexity in Software development projects: an empirical Assessment for simple thinking. Human Factors and Ergonomics in Manufacturing & Service Industries 21, 1–10 (2011)
32. Jones, C.: Estimating software costs. McGraw-Hill, Inc., Hightstown (1998)
33. Cheney, L.H.: Effects of individual characteristics, organizational factors and task characteristics on computer programmer productivity and job satisfaction. Information Management 7(4), 209–214 (1984)
34. Couger, J.D., Zawacki, R.A.: Motivating and managing computer personnel. Wileylnterscience, New York (1980)
35. Woodruff, C.K.: Data processing people - Are they really different? Information & Management 3(4), 133–139 (1980)

36. Beecham, S., Baddoo, N., Hall, T., Robinson, H., Sharp, H.: Motivation in Software Engineering: A systematic literature review. Information and Software Technology 50(9-10), 860–878 (2008)
37. Griesser, J.W.: Motivation and information system professionals. Journal of Managerial Psychology 8(3), 21–30 (1993)
38. Guterl, F.: Spectrum/Harris poll - The job. IEEE Spectrum 21(6), 38 (1984)
39. Basili, V.R., Reiter Jr., R.W.: An Investigation of Human Factors in Software Development. Computer 12(12), 21–38 (1979)
40. Humphrey, W.: Psp(Sm): A Self-Improvement Process for Software Engineers, 1st edn. Addison-Wesley Professional, Reading (2005)
41. Ren-zuo, X., Ruo-feng, M., Li-na, L., Zhong-wei, X.: Human Factors Analysis in Software Engineering. Wuhan University Journal of Natural Sciences 9(1), 18–22 (2004)
42. Bach, J.: Enough About Process: What We Need Are Heroes. Computer 12(2), 96–98 (1995)
43. Bach, J.: What Software Reality Is Really About. Computer 32(12), 148–149 (1999)
44. Beaver, J.M., Schiavone, G.A.: The effects of development team skill on software product quality. SIGSOFT Software Engineering Notes 31(3), 1–5 (2006)
45. Zuser, W., Grechenig, T.: Reflecting Skills and Personality Internally as Means for Team Performance Improvement. In: Proceedings of the 16th Conference on Software Engineering Education and Training (CSEET 2003), pp. 234–241. IEEE Computer Society, Washington (2003)
46. Ting-Peng, L., Jiang, J., Klein, G.S., Liu, J.Y.-C.: Software Quality as Influenced by Informational Diversity, Task Conflict, and Learning in Project Teams. IEEE Transactions on Engineering Management 57(3), 477–487 (2010)
47. Hoegl, M., Gemuenden, H.G.: Teamwork Quality and the Success of Innovative Projects: A Theoretical Concept and Empirical Evidence. Organization Science 12(4), 435–444 (2001)
48. Wong, B., Bhatti, M.: The influence of team relationships on software quality. In: ICSE Workshop on Software Quality, WOSQ 2009, pp. 1–8 (2009)
49. Fernández, L., García, M.J.: Software engineering professionalism. Upgrade (4), 42–46 (2003)
50. Murugesan, S.: Attitude towards testing: a key contributor to software quality. In: Proceedings of First International Conference on Software Testing, Reliability and Quality Assurance, pp. 111–115 (1994)
51. Acuña, S.T., Gómez, M., Juristo, N.: Towards understanding the relationship between team climate and software quality–a quasi-experimental study. Empirical Software Engineering 13(4), 401–434 (2008)
52. Gill, N.S.: Factors affecting effective software quality management revisited. SIGSOFT Software Engineering Notes 30(2), 1–4 (2005)
53. Fernández, L.: Personal Skills for Computing Professionals. Computer 42(10), 110–112 (2009)
54. Curtis, B., Hefley, W.E., Miller, S.: People Capability Maturity Model, Software Engineering Institute, Carnegie Mellon University (1995)
55. Curtis, B., Hefley, W.E., Miller, S.A.: People Capability Maturity Model: Guidelines for Improving the Work Force. Addison-Wesley Longman, Amsterdam (2002)
56. Curtis, B., Hefley, W.E., Miller, S.A.: Experiences Applying the People Capability Maturity Model. CROSSTALK The Journal of Defense Software Engineering, 9–13 (April 2003)

57. Fernández-Sanz, L., Villalba, M.T., Hilera, J.R., Lacuesta, R.: Factors with negative influence on software testing practice in spain: A survey. In: O'Connor, R.V., Baddoo, N., Cuadrago Gallego, J., Rejas Muslera, R., Smolander, K., Messnarz, R. (eds.) EuroSPI 2009. Communications in Computer and Information Science, vol. 42, pp. 1–12. Springer, Heidelberg (2009)
58. Lara, P.J., Fernández-Sanz, L.: Un experimento sobre hábitos de pruebas artesanales de software: Resultados y Conclusiones, Actas de Talleres de Ingeniería del Software y Bases de Datos. In: PRIS 2007: Taller sobre Pruebas en Ingeniería del Software, pp. 23–30 (2007)
59. Galin, D.: Software Quality Assurance: From Theory to Implementation. Addison-Wesley, Reading (2004)
60. Stewart, R., Priven, L.: How to Avoid Software Inspection Failure and Achieve Ongoing Benefits. CROSS TALK. The Journal of Defense Software Engineering 23-28 (January 2008)

A Framework for Classifying and Comparing Software Architecture Tools for Quality Evaluation

Eudisley Anjos and Mário Zenha-Rela

CISUC, Centre for Informatics and Systems
University of Coimbra, Portugal
{eudis,mzrela}@dei.uc.pt

Abstract. Software quality is a crucial factor for system success. Several tools have been proposed to support the evaluation and comparison of software architecture designs. However, the diversity in focus, approaches, interfaces and results leaves the researcher and practitioner wondering what would be the most appropriate solution for their specific goals. This paper presents a comparison framework that identifies the most relevant features for categorizing different architecture evaluation tools according to six different dimensions. The results show the attributes that a comprehensive tool should support include: the ability to handle multiple modelling approaches, integration with the industry standard UML or specific ADL, support for trade-off analysis of competing quality attributes and, the reuse of knowledge through the build-up of new architectural patterns. This comparison is able to, not only guide the choice of evaluation, but also promote the development of more powerful tools for modeling and analysis of software architectures.

Keywords: software quality challenges; trends for achieving quality objectives; architecture evaluation process; **s**oftware architecture; Affidavit.

1 Introduction

In the last years we have witnessed a growing interest in software architecture design driven by the unavoidable need to apply it in the development of large and complex software systems [1], [2]. The design of Software Architecture (SwA) is one of the earliest stages of system development. Ensuring that all requirements are fulfilled leads to a better implementation of the system [3]. However, ensuring the correctness of software architecture is not simple. Several tools and methods have been proposed to help the architects design and evaluate the system requirements right from an early architecture model.

System requirements are collected from stakeholders (e.g. end-users, management, marketing, third-parties, and developers) and include functional and nonfunctional attributes. Functional requirements typically describe the anticipated behavior of a solution or solution components, and a huge body of knowledge and literature on how to collect, model, evaluate and verify them exist [4], [5]. Non-functional attributes, on the other hand describe properties of the system e.g. maintainability, usability,

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 270–282, 2011.

usability, efficiency, portability and reliability, etc [6]. There is no consensus about a standard methodology on how to assess quality in SwA, and the lack of a common taxonomy (despite the IEEE std. 1061-1998) also constrains the development of widespread evaluation methods and tools. Moreover, architects can define specific attributes based on their context, which implies that the supporting methodologies are abstract enough to fit into their specific architecture design. So, the architect is expected to understand the requirements of stakeholders, assess the quality attributes, and choose the best evaluation method and tool to fit in the design process.

There are several methods in literature to evaluate software architecture quality namely: SAAM, Software Architecture Analysis Method [1], [7], with a focus on modifiability; ATAM, Architecture Trade-off Analysis Method [1], [8], mainly used to modifiability but also applied to other quality attributes evaluation and trade-off verification; ARID, Active Reviews for Intermediate Design [9], CBAM, Cost Benefits Analysis Method [10], FAAM, Family Architecture Assessment Method [11]. Most of these methodologies describe a set of steps that should be followed in order to check and evaluate the quality of the architectural design.

There are an increasing number of alternative approaches all of which make the architect's choice more difficult. Some researchers have proposed a taxonomy framework to classify and compare software architecture evaluation techniques, selecting key features of each methodology to categorize them. In [12] is proposed a framework to compare and assess eight SA evaluation methods (most of the scenario-based) and demonstrate the use of the framework to discover the similarities and differences among these methods. In [13] the authors describe the main activities in model checking techniques defining a classification and comparison framework and, in [14], the focus is on evaluation of ten models to assess performance, maintainability, testability and portability. These works mentioned before focus essentially on model checking, simulation-based or scenario-based approaches.

The architect can use evaluation tools (such as the tools evaluated in section 4 below) to support the application of methods, reducing the time spent on analysis and improving the results of the evaluation process. These tools perform different types of evaluation depending on the method used and which design features are the focus of assessment. Many architecture evaluation tools have been proposed but most of them are limited to a specific purpose (e.g. for deployment only) or propose a generic approach but developing only a subset of functionalities for demonstration purposes. Still, there is a lack of knowledge about what the relevant features should be addressed (e.g. description language used by the tool, quality parameters assessed, evaluation method, etc.). This makes difficult to choose the tool that best suits the architectural design process.

Some researchers have tried to assess architecture tools based on the evaluation techniques adopted and comparing them across different tools. However, little work has been done to classify and compare evaluation tools from a generic perspective, describing the main characteristics and assisting the perception of which are the relevant features these tools should provide.

The purpose of this paper is to present a comparison framework that enlights the most relevant attributes to help categorize the different architecture evaluation tools. We think this goal is relevant to the researchers and practitioners in the area as there is a growing number of methods and tools and it is easy to get overwhelmed by the seemingly diversity of options, such as ArchKriti [15], used to support crucial steps in architecture-based software development and, Attribute Driven Design method (ADD) [16], used to design software architecture of a system or collection of systems based on an explicit articulation of the quality attribute goals for the system. Moreover, since we are involved in a major international effort (EU/USA) to build a comprehensive architectural development support tool (AFFIDAVIT) [17], it is relevant to organize the current approaches to identify their major strengths and learn from their weaknesses. We do think that proper tools based on sound methodologies will bring us closer to the widespread adoption of tools that support architectural design, thus leading to an increase in the quality of software.

The reminder of the paper is organized as follows. In section 2 we present the fundamental role of quality attributes in the definition of software architecture and in section 3 the current software architecture evaluation categories. In the following section we discuss the selected architecture evaluation tools and which attributes were considered most relevant. In section 5 we compare the selected tools according to the identified relevant attributes and present the major insights and the current work on the Affidavit tool. Section 6 closes the paper by presenting some future work.

2 Quality Attributes and Software Architecture

The non-functional requirements of a Software Architecture play a key role in terms of the specifying the design patterns and deployment architectures of the solution being built [18]. For example, poor network connectivity will dictate requirements in terms of a centralized vs. decentralized deployment, requirements for resilience may dictate a requirement for clustered hardware, etc. To implement these architectural decisions and thus to design the whole architecture it is essential that the architect knows its most relevant quality attributes (e.g. modifiability, performance, security, availability.).

It is significant to consider that the two most important sources of quality attributes and requirements are the stakeholder's needs and the application domain [19], [20]. The architect needs to have a good understanding of these sources and their influence on the system to model the correct architecture to support the requirements. However, depending on the type of software and its context, different kinds of systems have different relevant quality attributes. In a complex long-lived application such as data-mining and network analysis, maintainability might be the most significant quality attribute while in a real-time sophisticated network, performance may drive most of the architectural decisions.

It is necessary to ensure the fulfillment of the quality parameters, although they are contradictory quite often (e.g. security versus usability, performance versus maintainability) [21]. According to Barbacci [22] designers need to analyze trade-offs among multiple conflicting attributes to attend the user requirements. The goal is to evaluate the interrelationships among quality attributes to get a better overall system (the best compromise for a given context). In order to help this kind of analysis, most of the evaluation methods include some sort of trade-off analysis.

So, evaluating the quality of a software architecture is a critical step to ensure that the software meets its design objectives in a reliable and predictable fashion. However the architect needs support to perform this activity. It is almost impossible for an architect validate manually the whole system model, verifying if the required quality attributes are effectively supported by the architecture and assuring that there are no conflicts between them. That is why the adoption of several methods, tools and ADLs are fundamental to help the architect in designing, specification and analysis of SwA.

3 Architecture Evaluation Methods

Any serious software architecture evaluation process needs to consider and categorize several different architectural aspects of the system's requirements (e.g. kind of requirements, architectural description, etc.). Depending on how these aspects are addressed by the evaluation methods, it is possible to identify different evaluation methods categories. Regardless of category, the evaluation methods can be used in isolation, but it is also possible and common to combine different methods to improve the insight and confidence in the architectural solution to evaluate different aspects of the software architecture, if needed. After deciding for a specific evaluation method, the architect has to select the Architecture Description Language (ADL), tool and the best suited technique to her or his specific project.

In this work is adopted the architecture evaluation methods' categorization carried in [14] and [23]. The authors consider four evaluation categories: *scenario-based*, *formal-modeling*, *experience-based* and *simulation* based. Other authors (e.g. [13]), use model-checking to address techniques which verify whether architectural specifications conform to the expected properties. Since in this paper we consider that all presented categories are used to in assess the architectural model, model-checking is not inserted as a different evaluation category.

Below we present a brief characterization of each category:

- **Scenario-Based:** Methods in this category use operational scenarios that describe the requirements to evaluate the system quality. The scenarios are used to validate the software architecture using architectural tactics and the results are documented for later analysis including system evolution, maintenance and the creation of a product line. There are several scenario-based evaluation methods namely SAAM [1], [7], ATAM [1], [8], CBAM [10], FAAM [11].
- **Formal-Modeling:** Uses mathematical proofs for evaluating the quality attributes. The main focus of this category is the evaluation of operational parameters such as performance. An example of formal-modeling is NOAM (Net-based and Object-based Architectural Model) [24]. Usually, the use of formal-modeling and simulation-based methods can be joined to estimate the fulfillment of specific qualities.
- **Experience-based:** The methods in this category use previous experience of architects and stakeholders to evaluate the software architecture [25]. The knowledge obtained of previous evaluations is maintained as successful examples to design new similar solutions and drive further architecture refinements.

- **Simulation-based:** Uses architectural component implementations to simulate the system requirements and evaluate the quality attributes. The methods in this category can be combined with prototyping to validate the architecture in the same environment of the final system. Examples include LQN[26] and RAPIDE [27].

It is important to notice that the evaluation categories are not directly linked to the evaluation tools. They specify how to apply the theory behind the tools and commonly define steps to assess the SwA quality. Some tools support different methods to get a better insight. In fact, very few ADLs, like Aesop [28], Unicon [29] and ACME [30], provide support for different evaluation processes. They are closely serving as an evaluation tool themselves and assist in modeling specific concepts of architectural patterns, although unfortunately in most cases these are applicable to restricted purposes only.

4 Architecture Evaluation Tools

The diversity of techniques focusing on restricted contexts and attributes turns the selection of an architectural method into a complex task. Architects typically need to adapt several models and languages depending on the attributes they want to evaluate. Thus, it is necessary to know different description languages, scenarios specification, simulation process, application contexts and others method's features to make the best choice and perform the intended evaluation process. While generic tools do not become widespread, we have observed that architects tend to adopt the methods, tools and ADL that they have previously been in contact with.

Many evaluation methods (e.g. ATAM), describe a sequence of manual activities that the architect should perform to identify the main issues concerning quality assessment. Based on these descriptions or instructions, software tools are used to automate only parts of the evaluation process, such as: architectural scenario definition, analysis of architectural components relationships and others. Automating the whole validation of architectural quality requires the mediation of the architect to tailor the model according to the requirements and to solve errors and conflicts detected by the tools.

ADLs also evolved and new features were assembled to aid the architect. According to [31] the tools provided by an ADL are the canonical components (referred also as *ADL toolkit* [32]). Among these components we can mention *active specification,* a tool that guides the architect or even suggests wrong behaviors in the design and, *architectural analysis,* which is the evaluation of the system properties detecting errors in the architectural phase and reducing costs of corrections during the development process. We have take into account that most tools have requirements that hamper the use of ADL already known by the architect, forcing the learning of new architecture description languages or requiring design the model directly at the interface of the tool.

Many characteristics have to be managed and balanced by the architect, thus selecting the best tool is not a simple task. Sometimes, it is necessary that the architect knows how the tool works to decide whether is useful in a specific project. The system requirements guide the architect about the type of tool to be used but still lack of specific information about methods and features or this information is dispersed, preventing a sound and informed *a-priori* evaluation by the architect.

There are few studies to assist the architect in the selection of the best tool to support the architectural quality evaluation. In [33] for example, the authors compare different knowledge management tools to understand SwA knowledge engineering and future research trends in this field. The knowledge about how to assess a particular kind of system requires that the architect knows what characteristics the tool should have.

In this paper we present a survey of different types of architecture evaluation tools and classify them according to the six dimensions presented in Table 1 below. The goal is to identify significant dimensions to analyze the applicability of an evaluation tool in a particular context to a specific goal.

Table 1. Dimensions used for the evaluation of tools

Feature	Description
Method	The evaluation method used. One tool can support several methods.
ADL	Assess if the tool has its own ADL, use another know ADL or if the architect must describe the architecture manually using the tool interface.
Qualities	Indicate which quality attributes are covered by the tool.
Trade-off	The ability of the tool to understand and measure the trade-offs between two or more quality attributes.
Stakeholders	Indicate if the tool supports the participation of stakeholders during the architecture evaluation or somewhere during the architecture design.
Knowledge	Evaluate if the tool preserves the knowledge (e.g. architectural patterns) since the last architectural evaluation for performing new designs.

5 Assessment of Tools

During this work we carried out an extensive survey and analysis of the literature to identify the most relevant tools and understand their focus. Afterward we selected five tools as being more representative and identified six dimensions to present what we think is the essence of the current results. In this section is presented an assessment of these tools using the selected dimensions.

We tried to cover the different methods of evaluation supported by the selected tools. However, it is important to understand that several tools use multiple methods to achieve the proposed objectives. This is especially true when the tool is more generic and supports different techniques according to the attribute which is under evaluation. The tools selected as most relevant due to their maturity and impact in the literature (most are based on research work, even if used in industry) are ArchE design assistant [34], Architecture Evaluation Tool (AET) [35], Acme Simulator [36], ArcheOpterix [37] and DeSi [38]. We shall now present a brief description of each tool, as well as a table summarizing their profile according with the dimensions selected.

5.1 ArchE Design Assistant

This tool is an Eclipse plug-in that manages reasoning frameworks (RF) to evaluate software architectures. The evaluation models are the knowledge sources and the Architecture Expert (ArchE) baseline tool manages their interaction. A relevant point of this *assistant* is that a researcher can concentrate on the modeling and implementation of a reasoning framework for the quality attribute of interest. The tool has no semantic knowledge and consequently supports any reasoning framework. So, ArchE is an assistant to explore quality-driven architecture solutions.

The ArchE receives from each RF a *manifesto*. The *manifesto* is a XML file that lists the element types, scenarios and structural information handled by the reasoning framework. In ArchE many actors collaborate to produce the solution of a problem. In this case, the actors are the RF and every other actor can read information provided by other RFs. This communication can produce new information useful to some of them.

The flexibility of ArchE is its major strength but also its major weakness: researchers and practitioners are able/forced to develop or adapt their own quality-attribute model if not already supported. An input conversion for non supported ADLs might also be necessary.

As the ArchE authors state:

"*ArchE is not intended to perform an exhaustive or optimal search in the design space; rather, it is an assistant to the architect that can point out "good directions" in that space.*"

Note that ArchE is most useful during the assessment phase of architectural development.

Table 2. Dimensions evaluated for ArchE

Dimensions	ArchE behaviour
Method	This tool uses scenarios-based method but each RF can have its own evaluation method.
ADL	There is no specific ADL linked with this tool, only the XML file used as *manifesto*.
Qualities	All quality attributes can be evaluated, depending which RF is used
Trade-Off	The trade-off among different RFs uses a 'traffic-light' metaphor to indicate potential scenario improvements when applying different tactics.
Stakeholders	It is an architect-focused tool; other stakeholders are only involved when scenarios are identified.
Knowledge	ArchE does not build knowledge from past evaluation (architectural patterns) to apply in new projects.

5.2 Architecture Evaluation Tool (AET)

AET is a research tool developed at BOSCH to support the evaluation team in documenting results and managing information during an architecture review. AET has two databases to assist the architect with the information management: General, (i.e. static data) and Project (i.e. dynamic data) databases. This tool is present during the gathering of quality attributes information and architectural scenarios. So, the information obtained from stakeholders is stored in AET databases. The information is used in the current project and storage in the general database for new architectural projects. Although this tool was initially developed to evaluate performance and security, the project is still under development to include more attributes. This tool is focused in the initial phases of requirements gathering and quality attributes trade-off analysis.

Table 3. Dimensions evaluated for AET

Dimensions	AET behaviour
Method	This tool use scenarios as main method for evaluation. It uses both dynamic and static (experience-based) evaluation types.
ADL	There is no ADL linked with this tool. The data is inserted directly using the tool interface.
Qualities	All quality attributes can be evaluated as it is a mostly manual processing.
Trade-Off	The trade-off is performed based on the data introduced by the architect and stakeholders during the achievement of quality attributes and scenarios. The tool combines this information to guide the architect showing the risks and the impact of changes.
Stakeholders	In the initial steps of data gathering, the tool requires that stakeholders participate to fill the quality requirements (scenarios).
Knowledge	The knowledge (experience repository) is stored in database as input to new evaluations.

5.3 Acme Simulator

This tool is an extension of AcmeStudio (a plug-in of the Eclipse Framework) and uses its existing features for defining architectural models. Also provides specific architectural styles to specify relevant properties and topology to the kind of analysis. The Acme simulator as originally developed provides security and performance analysis using Monte Carlo simulation to evaluate the properties under certain assumptions about their stochastic behaviour. Since the simulator is written in AcmeStudio framework, the Eclipse allows flexible extensions and the tool uses Acme as design ADL to model the software architectures. This tool is clearly focused on architectural assessment.

5.4 ArcheOpterix

This tool provides a framework to implement evaluation techniques and optimization heuristics for AADL (Architecture Analysis and Description Language) based specifications. The algorithms should follow the principle of model-driven engineering, allowing reason about quality attributes based on an abstract architecture model. The focus of ArcheOpterix is embedded and pervasive systems.

The quality evaluation is represented using AttributeEvaluator modules that implements an evaluation method and provides metrics for a given architecture. This tool can evaluate all quality attributes as long as there are suitable evaluation algorithms. The two initial attribute evaluators use mathematical methods to measure the goodness of a given deployment, data transmission reliability and communication overhead. Thus, this tool was classified as formal, although the AttributeEvaluator may use different methods.

Table 4. Dimensions evaluated for Acme Simulator

Dimensions	Acme Simulator behaviour
Method	Essentially a Monte-Carlo simulation using specifically designed scenarios (behavior model trees) for evaluation.
ADL	Acme Simulator is linked with AcmeStudio and Acme ADL is necessary to model the architecture design before the evaluation.
Qualities	Initially performance and security were evaluated using a Monte Carlo simulation. The approach is general enough to be extended to other quality attributes.
Trade-Off	The trade-off is presented as a table; the comparison is realized manually with the information provided by the tool. The authors plan to support the comparison directly.
Stakeholders	The stakeholders do not participate during the use of this tool.
Knowledge	No knowledge is explicitly preserved to new projects.

Table 5. Dimensions evaluated for ArcheOpterix

Dimensions	ArcheOpterix behaviour
Method	Although the tool uses formal modeling to evaluate the attributes, the *AttributeEvaluator* which contains the algorithm can adopt other methods.
ADL	Uses the AADL standard to describe the architecture to be evaluated.
Qualities	All quality attributes can be evaluated as long an algorithm exists. Currently two quality attributes have been evaluated: data transmission reliability and communication overhead.
Trade-Off	The tool uses an architecture optimization module to solve multi-objective optimization problems using evolutionary algorithms.
Stakeholders	The stakeholders do not participate using this tool.
Knowledge	ArcheOpterix does not preserve the knowledge of evaluations to be used in new projects.

5.5 DeSi

DeSi is an environment that supports flexible and tailored specification, manipulation visualization and re-estimation of deployment architectures for large-scale and highly distributed systems. Using this tool it is possible to integrate, evaluate and compare different algorithms improving system availability in terms of feasibility, efficiency and precision.

This tool was implemented in the Eclipse platform. Its architecture is flexible enough to allow exploration to other system characteristics (e.g., security, fault-tolerance, etc.). DeSi was inspired in tools for visualizing system deployment as UML improving support to specifying, visualizing and analyzing different factors that influence the quality of a deployment.

Table 6. Dimensions evaluated for DeSi

Dimensions	DeSi behaviour
Method	Formal, DeSi uses algorithms to improve system availability
ADL	The data is input directly in DeSi interface and the tool does not use any ADL.
Qualities	Availability. Although depending on the taxonomy, some features could be classified as security and performance. Moreover, the tool allows the use of new algorithms to explore different attributes.
Trade-Off	There is no trade-off function. DeSi simply provides a benchmarking capability to compare the performance of various algorithms.
Stakeholders	Stakeholders participation is not present.
Knowledge	No knowledge is preserved for future evaluations.

5.6 General Overview

In the Table 7, we assembled a summary of results for the selected tools. A rapid analysis provides a synthesis of the most relevant dimensions that are required to assess its nature, applicability and usefulness for a specific goal.

As can be seen, although every tool claims to be devoted to architectural assessment their focus can be very distinct. While ArchE, AET and ArcheOpterix goals cover every quality attribute (depending on whether the corresponding models are implemented), AcmeSimulator and DeSi have a specific focus. On the other hand, while ArchE, AET and AcmeSimulator support human guided trade-off analysis, Archepterix and DeSi provide absolute assessment metrics. As such there is no 'best' tool, as it depends on the usage envised by the architect to face a specific problem in a specific context.

6 Conclusions and Further Work

It is very difficult to classify and compare architectural assessment tools that use different methods and techniques. The large amount of features and attributes that can be questioned hampers this process and demonstrate that generic tools that can be tailored to the architect needs are required. The more flexibility and features available the more it is customizable to the specific architect needs.

Using the assessment insight provided by our approach (Table 7) is possible not only improve the selection of the more appropriate tool for a specific context, but also serve as a powerful driver for building an advanced integrated tool to support architectural modeling and analysis as it shows the attributes that such a comprehensive tool should support. The features presented here are essentially: the ability to handle multiple modelling approaches, integration with the industry standard UML, but also specific ADL whenever needed, support of trade-off analysis of multiple competing quality attributes and, of utmost importance, the possibility to reuse knowledge through the reuse and build-up of new architectural solutions.

This is part of our contribution to a major international effort (USA/EU) in developing a tool (AFFIDAVIT) to promote the widespread adoption of sound architectural practices in the software engineering community, consequently, aiming to increase global software quality. Our specific focus is on maintainability addressing methods to assess this attribute in software architectures, and metrics to assess the level of maintenance during the architectural design.

As future work we intend to improve this analysis including with more dimensions (e.g. application domain, flexibility, open-source, etc.) and publish this framework on Web to be used by researchers and practitioners. Furthermore, in the context of our work, an evaluation of maintainability properties assessed by these tools should be done.

Table 7. Overview of tools according to the relevant dimensions

	ArchE	AET	Acme Simulator	ArcheOpterix	DeSi	*Affidavit*
Method	Scenarios	Scenarios	Simulation	Formal	Formal	All
ADL			✓	✓		✓
Q.A.	All	All	Performance Security	All	Availability	All
Trade-off	✓	✓	✓			✓
Stakeholders		✓				✓
Knowledge		✓				✓

References

[1] Clements, P., Kazman, R., Klein, M.: Evaluating Software Architectures: Methods and Case Studies. Addison-Wesley Professional, Reading (2001)

[2] Yang, H., Ward, D.M.: Successful evolution of software systems. Artech House, Boston (2003)

[3] Pressman, R., Pressman, R.: Software Engineering: A Practitioner's Approach, 6th edn. McGraw-Hill Science/Engineering/Math., New York (2004)

[4] Hausmann, J.H., Heckel, R.: Detection of Conflicting Functional Requirements in a Use Case-Driven Approach - A static analysis technique based on graph transformation. In: ICSE 2002, pp. 105–115 (2002)
[5] Post, H., Sinz, C., Merz, F., Gorges, T., Kropf, T.: Linking Functional Requirements and Software Verification. In: IEEE International Conference on Requirements Engineering, pp. 295–302 (2009)
[6] Chung, L., do Prado Leite, J.C.S.: On non-functional requirements in software engineering. In: Borgida, A.T., Chaudhri, V.K., Giorgini, P., Yu, E.S. (eds.) Conceptual Modeling: Foundations and Applications. LNCS, vol. 5600, pp. 363–379. Springer, Heidelberg (2009)
[7] Kazman, R., Bass, L., Abowd, G., Webb, M.: SAAM: A Method for Analyzing the Properties of Software Architectures (1994)
[8] Barbacci, M.R., et al.: Steps in an Architecture Tradeoff Analysis Method: Quality Attribute Models and Analysis, Software Engineering Institute, Carnegie Mellon University, pp. 219–230 (1998)
[9] Clements, P.C.: Active Reviews for IntermediateDesigns. SEI: Carnegie Mellon University (2000)
[10] Klein, M., Kazman, R., Asundi, J.: Quantifying the costs and benefits of architectural decisions. In: Proceedings of the 23rd International Conference on Software Engineering, ICSE 2001, pp. 297–306 (2001)
[11] Dolan, W., Hammer: Technische Universiteit Eindhoven, Architecture assessment of information-system families: a practical perspective, Technische Universiteit Eindhoven (2001)
[12] Babar, M.A., Zhu, L., Jeffery, R.: A Framework for Classifying and Comparing Software Architecture Evaluation. In: Proceedings Australian Software Engineering Conference (ASWEC), vol. 2004, pp. 309–318 (2004)
[13] Zhang, P., Muccini, H., Li, B.: A classification and comparison of model checking software architecture techniques. Journal of Systems and Software 83(5), 723–744 (2010)
[14] Mattsson, M., Grahn, H., Mårtensson, F.: Software Architecture Evaluation Methods for Performance, Maintainability, Testability, and Portability. In: Presented at the Second International Conference on the Quality of Software Architectures, QoSA 2006 (2006)
[15] Vallieswaran, V., Menezes, B.: ArchKriti: A Software Architecture Based Design and Evaluation Tool Suite. In: Fourth International Conference on Information Technology, ITNG 2007, pp. 701–706 (2007)
[16] Bachmann, F., Bass, L.: Introduction to the attribute driven design method. In: Proceedings of the 23rd International Conference on Software Engineering, Washington, DC, USA, pp. 745–746 (2001)
[17] Affidavit Project (February 2011), http://affidavit.dei.uc.pt/
[18] Department, A.J., Jansen, A.: Software Architecture as a Set of Architectural Design Decisions. In: Proceedings of WICSA 2005, pp. 109–120 (2005)
[19] Al-Naeem, T., Gorton, I., Babar, M.A., Rabhi, F., Benatallah, B.: A quality-driven systematic approach for architecting distributed software applications. In: Proceedings of the 27th International Conference on Software Engineering, St. Louis, MO, USA, pp. 244–253 (2005)
[20] Niemelä, E.: Strategies of Product Family Architecture Development. In: Software Product Lines, pp. 186–197 (2005)
[21] Henningsson, K., Wohlin, C.: Understanding the Relations between Software Quality Attributes - A Survey Approach. Presented at the 12th International Conference for Software Quality, Ottawa - Canada (2002)

[22] Barbacci, M., Klein, M., Longstaff, T., Weinstock, C.: Quality Attributes: Technical Report. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA (1995)

[23] Bosch, J.: Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach. Addison-Wesley Professional, Reading (2000)

[24] Deng, Y., Lu, S., Evangelist, M.: A Formal Approach for Architectural Modeling and Prototyping of Distributed Real-Time Systems. Presented at the Thirtieth Hawaii International Conference Digital on System Sciences (1997)

[25] Rosso, C.: Continuous evolution through software architecture evaluation: a case study. Journal of Software Maintenance and Evolution: Research and Practice 18(5) (2006)

[26] Aquilani, F., Balsamo, S., Inverardi, P.: Performance analysis at the software architectural design level. Performance Evaluation 45(2-3), 147–178 (2001)

[27] Luckham, D.C., Kenney, J.J., Augustin, L.M., Vera, J., Bryan, D., Mann, W.: Specification and analysis of system architecture using Rapide. IEEE Transactions on Software Engineering 21(4), 336–354 (1995)

[28] Garlan, D., Allen, R., Ockerbloom, J.: Exploiting Style in Architectural Design Environments (1994)

[29] Shaw, M., DeLine, R., Klein, D.V., Ross, T.L., Young, D.M., Zelesnik, G.: Abstractions for Software Architecture and Tools to Support Them. IEEE Transactions on Software Engineering 21, 314–335 (1995)

[30] Garlan, D., Monroe, R., Wile, D.: Acme: an architecture description interchange language. In: CASCON First Decade High Impact Papers, New York, NY, USA, pp. 159–173 (2010)

[31] Medvidovic, N., Taylor, R.N.: A Classification and Comparison Framework for Software Architecture Description Languages. IEEE Trans. Softw. Eng. 26(1), 70–93 (2000)

[32] Garlan, D., Ockerbloom, J., Wile, D.: Towards an ADL Toolkit, presented at the EDCS Architecture and Generation Cluster (2008)

[33] Tang, A., Avgeriou, P., Jansen, A., Capilla, R., Ali Babar, M.: A comparative study of architecture knowledge management tools. Journal of Systems and Software 83(3), 352–370 (2010)

[34] Diaz-Pace, A., Kim, H., Bass, L., Bianco, P., Bachmann, F.: Integrating Quality-Attribute Reasoning Frameworks in the ArchE Design Assistant. In: Proceedings of the 4th International Conference on Quality of Software-Architectures: Models and Architectures, pp. 171–188 (2008)

[35] Thiel, S., Hein, A., Engelhardt, H., Bosch, R.: Tool Support for Scenario-Based Architecture Evaluation. Presented at the 25th International Conference on Software Engineering (2003)

[36] Schmerl, B., Butler, S., Garlan, D.: Architecture-based Simulation for Security and Performance (2006)

[37] Aleti, A., Bjornander, S., Grunske, L., Meedeniya, I.: ArcheOpterix: An extendable tool for architecture optimization of AADL models. In: International Workshop on Model-Based Methodologies for Pervasive and Embedded Software, Los Alamitos, CA, USA, pp. 61–71 (2009)

[38] Mikic-rakic, M., Malek, S., Beckman, N., Medvidovic, N.: A Tailorable Environment for Assessing the Quality of Deployment Architectures in Highly Distributed Settings. Presented at the Working Conf. on Component Deployment, pp. 1–17 (2004)

Causes of the Violation of Integrity Constraints for Supporting the Quality of Databases

Hendrik Decker*

Instituto Tecnológico de Informática, Valencia, Spain

Abstract. The quality of the information provided by databases can be captured by integrity constraints. Thus, violated cases of constraints may serve as a basis for measuring the quality of given database states. A quality metric with the potential of more accuracy is obtained by measuring the causes, i.e., data that are responsible for constraint violations. Such measures also serve for controlling quality impairment across updates.

1 Introduction

In [5,7], we have shown that the quality of stored data can be modeled, measured and monitored by declarative integrity constraints. The latter describe semantic properties of stored data that are required to hold in each database state. Thus, violations of such constraints reflect a lack of data quality. Hence, as soon as violations of integrity can be quantified, the quality of the data on which the constraints are imposed can be measured. Quality can then also be controlled, by checking, upon each update, whether the amount of integrity violation increases or decreases. We are going to see that most, though not all known integrity checking methods in the literature can be used for this kind of control.

Constraint violations can be quantified in several ways. In [5,7], the basic idea of quantifying the quality of stored data was to count the number of violated cases, i.e. instances of constraints that are not satisfied. Not only the cardinality of sets of violated cases, but also these sets themselves constitute a valuable metric space, together with the partial order of set inclusion.

In this paper, we show that, instead of *cases*, also the *causes* of violated cases can be counted, for measuring the amount of extant inconsistency in a database. And, similar to cases, also the powerset of causes of violations constitutes a metric space in which sets of causes can be compared for monitoring whether given updates would introduce new integrity violations or not. Causes essentially are minimal sets of stored data that are sufficient for violating some constraint. They are easily determined for denial constraints without negated literals, but more involved if there is non-monotonic negation.

While cases of violated constraints are only statements *about* the lack of quality of stored data, causes precisely *are* the data that lack quality. Hence, counting

* Partially supported by FEDER and the Spanish grants TIN2009-14460-C03 and TIN2010-17139.

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 283–292, 2011.

causes and checking their increase or decrease may provide more accurate quality metrics than cases. Essentially, the main idea of this paper is to improve the original concept of quantifying the quality of databases as elaborated in [5,7], by replacing cases with causes in the main definitions.

In Section 2, we outline the formal framework of this paper. In Section 3, we revisit the definition of cases, originally used for inconsistencytolerant integrity checking [6]. Cases are instances of constraints that are apt to model the quality of information in a more differentiated manner than universally quantified constraints. In Section 4, we discuss several inconsistency metrics. In particular, we revisit the metrics based on cases, as introduced in [5,7], and introduce metrics based on causes in an analogous manner. In Section 5, we show that, apart from providing quality metrics, measures of integrity violations based on cases or causes also may serve to control and contain impaired data quality across updates. In Section 6, an outlook to future work concludes the paper.

2 Background and Framework

Background and framework of this paper, including notations and terminology, are consitituted by *datalog*, as referred to in the standard literature, e.g., [1] [9].

2.1 Databases, Updates, Constraints

An *atom* is an expression of the form $p(a_1, ..., a_n)$, where p is a predicate of arity n ($n \geq 0$); the a_i, called *arguments*, are either constants or variables. A *literal* is either an atom A or a negated atom $\sim A$. A *fact* is an atom where all arguments are constants. A *database clause* is either a *fact*, the predicate of which corresponds to a relational table and the arguments of which correspond to column values in that table, or a formula of the form $A \leftarrow B$, where the *head* A is an atom and the *body* B is a conjunction of literals; all variables in $A \leftarrow B$ are implicitly quantified universally in front of the formula.

A *database* is a finite set of database clauses. For each database D and each predicate p, the set of clauses D_p in D the head of which has predicate p is called the *definition* of p in D. If q is a predicate in the body of some clause in D_p, then p is said to *recur* on q; p is also said to *recur* on each predicate on which q recurs. A database D is *definite* if there is no negated atom in the body of any clause in D. It is *hierarchical* if there is no predicate in D that recurs on itself.

An *update* is a finite set of database clauses to be inserted or deleted. For an update U of a database state D, we denote the updated database, where all inserts in U are added to D and all deletes in U are removed from D, by D^U.

An *integrity constraint* (in short, *constraint*) is a first-order predicate logic sentence which, for convenience and without loss of expressive power, we assume to be always represented as *denials*, i.e., formulas of the form $\leftarrow B$, where the body B states what must not hold. Implicitly, each variable in B is universally quantified at the front of $\leftarrow B$. A denial is *definite* if there is no negated atom in its body. An *integrity theory* is a finite set of constraints.

Constraints can be read as *necessary* conditions for the quality of stored data. If all intended application semantics are expressed by constraints, then the integrity theory represents a complete set of conditions that, when satisfied, is also *sufficient* for ensuring the quality of the stored data.

The DBMS is supposed to ensure that the database satisfies its integrity theory at all times, i.e., that all constraints are logical consequences of each state. To achieve this, database theory requires that, for each update U, the 'old' database D, i.e., the state to be updated by U, must satisfy all constraints, such that integrity checking can focus on those constraints that are possibly affected by the update. If those constraints remain satisfied, then the 'new' state D^U reached by committing U also satisfies all constraints.

Despite the theoretical insistence on total consistency at all times, the quality of a database is likely to suffer and deteriorate in the course of its evolution. Hence, it is necessary to have mechanisms that are able to tolerate certain amounts of integrity violations. As we are going to see, the cause-based approach developed in this paper is inconsistency-tolerant.

From now on, let D, IC, I, U and adornments thereof always stand for a database, an integrity theory, a constraint and, resp., an update. For convenience, we write $D(I) = true$ (resp., $D(I) = false$) if I is satisfied (resp., violated) in D. Similarly, $D(IC) = true$ (resp., $D(IC) = false$) means that all constraints in IC are satisfied in D (resp., at least one constraint in IC is violated in D).

2.2 Integrity Checking

The quality of stored data that is described by integrity constraints can be controlled by checking integrity for each update that could potentially violate it.

Below, we revisit a definition of integrity checking that abstracts away from any technical detail of how checking is done [6]. It describes each integrity checking method $\mathcal{M}$ as a mapping that takes as input a database D, and integrity theory IC and an update U, and outputs either *sat* or *vio*. If $\mathcal{M}$ is sound, $\mathcal{M}(D, IC, U) = sat$ indicates that U preserves integrity satisfaction, i.e., $D^U(IC) = true$. If $\mathcal{M}$ is also complete, then $\mathcal{M}(D, IC, U) = vio$ indicates that U violates integrity i.e., $D^U(IC) = false$. Also the output *vio* of an incomplete method may mean that the update would violate integrity. But it may also mean that further checking is needed for determining the integrity status of D^U; if there are not enough resources to do so, then U should be cautiously rejected.

Definition 1. (*Sound and complete integrity checking*)
Let $\mathcal{M}$ be a method for integrity checking. $\mathcal{M}$ is called *sound* or, resp., *complete* if, for each *(D,IC,U)* such that $D(IC)$=*true,* (1) or, resp., (2) holds.

$$\text{If } \mathcal{M}(D, IC, U) = sat \text{ then } D^U(IC) = true. \tag{1}$$

$$\text{If } D^U(IC) = true \text{ then } \mathcal{M}(D, IC, U) = sat. \tag{2}$$

Quality maintenance by integrity checking would tend to be too expensive, unless some simplification method were used [2]. Simplification essentially means that, for an update U, it suffices to check only those instances of constraints that are potentially violated by U. That idea is the basis of most methods for integrity checking proposed in the literature or used in practice [2].

Example 1. Let *married*(*Man*, *Woman*) be a relation with predicate *married* about married couples in the database of a civil registry (column names are self-explaining). Let I be the denial constraint

$$I \;=\; \leftarrow \mathit{married}(x,y),\, \mathit{married}(x,z),\, y \neq z\,.$$

I states that no man x may be married to two different women y and z, i.e., I forbids bigamy.

Now, let U be an update that inserts *married*(*joe*, *sue*). Usually, integrity then is checked by evaluating the following instance I' of I:

$$I' \;=\; \leftarrow \mathit{married}(\mathit{joe},\mathit{sue}) \wedge \mathit{married}(\mathit{joe},z) \wedge \mathit{sue} \neq z\,.$$

Since U makes *married*(*joe*, *sue*) true, I' can be simplified to

$$I'_s \;=\; \leftarrow \mathit{married}(\mathit{joe},z) \wedge \mathit{sue} \neq z\,.$$

The simplification I'_s asks if *joe* is married to any person z whose name is not *sue*. Its evaluation essentially amounts to a simple search in the table of *married*, in order to see if there is any woman with name other than *sue* who would be married with *joe*. If so, U is rejected; if not, U can be committed. Clearly, the evaluation of I' is significantly cheaper than the evaluation of I, which would involve a join of the entire *married* relation with itself.

In principle, also the following instance I'' of I would have to be evaluated:

$$I'' \;=\; \leftarrow (\mathit{married}(\mathit{joe},y) \wedge \mathit{married}(\mathit{joe},\mathit{sue})\,. \wedge y \neq \mathit{sue})$$

I'' is obtained by resolving *married*(x, z) in I with the update *married*(*joe*, *sue*). However, no evaluation of I'' is needed since I'' is logically equivalent to I'.

3 Causes of the Violation of Integrity

Informally, causes are minimal explanations of why a constraint is violated. Similarly, causes also may serve as concise justifications of why an answer to a query has been given, and for computing answers that have integrity in inconsistent databases, as shown in [4,3]. In this paper, we are going to use causes for two purposes: firstly, in Section 4, for quantifying the lack of quality in databases, by sizing sets of causes of integrity violation; secondly, in Section 5, for inconsistency-tolerant integrity checking, by which quality can be controlled.

For simplicity, we contend ourselves in this paper with causes of the violation of definite denials, i.e. constraints without negation in their bodies, and definite databases. Due to the non-monotonicity of database negation, the definition of causes for more general constraints in hierarchical databases is more complicated; it is elaborated in [3].

Definition 2. (*cause*)
Let D be a database and $I = \leftarrow B$ be a constraint.

a) A set E such that each element in E is a ground instance of a clause in D is called an *explanation* of the violation of I in D if there is a substitution θ such that $E \models B\theta$.

b) An explanation C of the violation of I in D is a *cause* of the violation of I in D if no explanation of the violation of I in D is a proper subset of C.

c) E is called a *cause* of the violation of IC in D if there is an $I \in IC$ of which E is a cause.

In part a, $E \models B\theta$ entails that $D \models B\theta$, i.e., I is violated in D. Parts b and c formalize that causes are minimal explanations of the violation of I or, resp., IC in D.

Example 2. Let the constraint $I = \leftarrow \mathit{married}(x, y), \mathit{same\text{-}sex}(x, y)$ be imposed on the database D of Example 1. The predicate *same-sex* be defined in D by $\mathit{same\text{-}sex}(x, y) \leftarrow \mathit{male}(x),\ \mathit{male}(y)$ and $\mathit{same\text{-}sex}(x, y) \leftarrow \mathit{female}(x),\ \mathit{female}(y))$. Clearly, I denies cosexual marriages. Yet, assume $\mathit{married}(\mathit{fred}, \mathit{rory})$ is a fact in D, where both $\mathit{male}(\mathit{fred})$ and $\mathit{male}(\mathit{rory})$ hold, the latter due to a registered gender reversal after marriage. Formally, that amounts to a violation of I. Thus, $\{\mathit{married}(\mathit{fred}, \mathit{rory}), \mathit{male}(\mathit{fred}), \mathit{male}(\mathit{rory})\}$ is a cause of the violation of I in D.

It is easy to see that causes can be obtained as a by-product of standard query evaluation, i.e., the computation of causes is virtually for free.

4 Measuring Quality by Quantifying Causes

Database quality can be quantified by measuring inconsistency, and in particular by quantifying causes of integrity violations. In 4.1, we present a general axiomatization of inconsistency metrics. It enhances the axiomatization in [5,7]. In 4.2, we introduce a quality metric based on causes. In 4.3, we argue why cause-based inconsistency metrics are preferable to case-based metrics.

4.1 Axiomatizing Inconsistency Metrics

Let $\preccurlyeq$ symbolize an ordering that is antisymmetric, reflexive and transitive. For expressions E, E', let $E \prec E'$ denote that $E \preccurlyeq E'$ and $E \neq E'$. Further, for two elements A, B in a lattice, let $A \oplus B$ denote their least upper bound.

Definition 3. We say that $(\mu, \preccurlyeq)$ is an *inconsistency metric* (in short, a *metric*) if μ maps pairs (D, IC) to some lattice that is partially ordered by $\preccurlyeq$, and, for each pair (D, IC) and each pair (D', IC'), the following properties (3) – (6) hold.

$$\text{If } \ D(IC) = \mathit{true} \ \text{ and } \ D'(IC') = \mathit{false} \ \text{ then } \ \mu(D, IC) \prec \mu(D', IC') \tag{3}$$

$$\text{If } \ D(IC) = \mathit{true} \ \text{ then } \ \mu(D, IC) \preccurlyeq \mu(D', IC') \tag{4}$$

$$\mu(D, IC \cup IC') \ \preccurlyeq \ \mu(D, IC) \oplus \mu(D, IC') \tag{5}$$

$$\mu(D, IC) \ \preccurlyeq \ \mu(D,\ IC \cup IC') \tag{6}$$

Property (3), called *violation is bad* in [5,7], ensures that the measured amount of inconsistency in any pair (D, IC) for which integrity is satisfied is always smaller than what is measured for any pair (D', IC') for which integrity is violated. Property (4), called *satisfaction is best*, ensures that inconsistency is lowest, and hence quality is always highest, in any database that totally satisfies its integrity theory. Property (5) is a triangle inequality which states that the inconsistency of a composed element (i.e., the union of (D, IC) and $(D, IC'))$ is never greater than the least upper bound of the inconsistency of the components. Property (6) requires that the values of μ grow monotonically with growing integrity theories.

Occasionally, we may identify a metric $(\mu, \preccurlyeq)$ with μ, if $\preccurlyeq$ is understood.

Example 3. A simple example of a coarse, binary inconsistency metric β is provided by the equation $\beta(D, IC) = D(IC)$, with the natural ordering $true \prec false$ of the range of β, i.e., integrity satisfaction $(D(IC) = true)$ means lower inconsistency and hence higher quality than integrity violation $(D(IC) = false)$.

More interesting inconsistency metrics are defined and featured in [5,7]. In fact, property (6) of Definition 3 has not been discussed in [5,7], but all examples of inconsistency metrics given there actually satisfy (6) as well.

For instance, the function that maps pairs (D, IC) to the cardinality of the set of cases (instances) of violated constraints is a convenient quality metrics. Inconsistency can also be measured by taking such sets themselves, as elements of the metric set that is constituted by the powerset of all cases of IC, together with the subset ordering. Other metrics can be based on causes of violations, as outlined in the following subsection.

4.2 Cause-Based Inconsistency Metrics

The lack of quality in databases can be reflected by counting and comparing sets of causes of the violation of constraints.

Let $\mathsf{CauVio}(D, I)$ denote the set of causes of the violation of I in D, and $\sigma(D, IC) = \{C \mid C \in \mathsf{CauVio}(D,I),\, I \in IC\}$ be the set of all causes of the violation of any constraint in IC. Then, $(\sigma, \subseteq)$ is an inconsistency metric, and so is $(\zeta, \leq)$, where the mapping ζ is defined by $\zeta(D, IC) = |\sigma(D,\, IC)|$ and $|\,.\,|$ denotes set cardinality. In words, ζ counts causes of integrity violation.

Example 4. Let IC consist of the constraint I (no bigamy) in Example 1, and a person named *sheik* be registered as a male citizen. i.e., $male(sheik) \in D$. Further, suppose that also the n facts $married(sheik,\, wife_1), \ldots, married(sheik,\, wife_n)$ $(n \geq 2)$ are in D, and that there is no other man in D who is married more than once. Then, for each i, j such that $1 \leq i, j \leq n$ and $i \neq j$, $\{married(sheik,\, wife_i)$, $married(sheik,\, wife_j)\}$ is a cause of the violation of IC in D. Hence, the inconsistency in D as measured by ζ is $\zeta(D, IC) = 1 + 2 + \ldots + n$–1. Thus, for $n > 3$, the inconsistency as measured by ζ that is caused by a man who is married to n different women is higher than the inconsistency of n men being married to just 2 women.

4.3 Causes vs Cases

As seen in Example 3, integrity and hence the quality of databases can be coarsely measured by checking the integrity constraints imposed on them. In [5,7], we have shown that this assessment of quality can be further refined by focusing on cases, i.e., instances of constraints that are actually violated. That focus takes into account that the majority of instances of constraints (which typically are universally quantified formulas, in general) remains satisfied, while only certain cases lack integrity and hence suffer from quality impairment.

Still, the association of the quality of a database with constraint violations, or even with violated cases of constraints, does not directly tell which are the actually stored data that are responsible for violating constraints, i.e. for the lack of quality. Hence, measuring quality by quantifying causes is preferable to the case-based approach, as illustrated by the following example.

Example 5. Suppose the predicate p in the constraint $I = \leftarrow p(x, x)$ (which requires the relation corresponding to p to be anti-reflexive) is defined by the two clauses $p(x, y) \leftarrow q(x, y), q(y, x)$ and $p(x, y) \leftarrow r(x, z), s(y, z)$. Further, suppose that the case $I' = \leftarrow p(c, c)$ of I is violated. With that information alone, as provided by focusing on violated constraints, it is not clear whether the violation of I is due to the existence of the tuple $q(c, c)$ in the database or to the existence of one or several pairs of tuples of the form $r(c, z)$ and $s(c, z)$ in the join of r and s on their respective last column. In fact, an arbitrary number of causes for the violation of I and even of I' may exists, but the case-based approach of quantifying the quality of databases does not give any account of that. As opposed to that, the cause-based approach presented in 4.2 clearly does.

Another advantage of causes over cases is that the latter do not provide any means for computing reliable answers to queries in inconsistent databases, while the former do, as shown in [4,3].

5 Controlling Impaired Quality

In 5.1, we are going to recapitulate from [5,7] that metric-based inconsistency-tolerant integrity checking can be used to monitor and control the evolution of impaired quality across updates. In 5.2, we are going to define inconsistency-tolerant cause-based integrity checking methods, by which the monitoring and control of quality impairment can be implemented.

5.1 Metric-Based Inconsistency-Tolerant Integrity Checking

For monitoring and controlling quality impairment in databases, it is desirable to have a mechanism that is able to preserve or improve the quality across updates, while tolerating extant impairments of quality. By Definition 4, below, which generalizes Definition 1, we are going to see that each inconsistency metric induces a sound integrity checking method that provides the desired properies.

Definition 4. (*metric-based inconsistency-tolerant integrity checking*)
Let $\mathcal{M}$ be a method for integrity checking and $(\mu, \preccurlyeq)$ be an inconsistency metric. $\mathcal{M}$ is called *sound*, resp., *complete wrt. metric-based inconsistency tolerance* if, for each triple (D, IC, U), (7) or, resp., (8) holds.

$$\text{If } \mathcal{M}(D, IC, U) = sat \text{ then } \mu(D^U, IC) \preccurlyeq \mu(D, IC). \tag{7}$$

$$\text{If } \mu(D^U, IC) \preccurlyeq \mu(D, IC) \text{ then } \mathcal{M}(D, IC, U) = sat. \tag{8}$$

If (7) holds, then $\mathcal{M}$ is also called *metric-based*, and, in particular, *μ-based*.

Definitions 1 and 4 are structually quite similar. However, there are two essential diferences. Firstly, the premise $D(IC) = true$ in Definition 1 is missing in Definition 4. This premise requires that integrity be totally satisfied before the update U. By contrast, inconsistency-tolerant integrity checking, as characterized by Definition 4, does not expect the total satisfaction of all integrity constraints. Rather, it ignores any extant violations (since the total integrity premise is absent), but prevents that the quality degrades across updates by additional violations, as guaranteed by the consequence of condition (7). So, the second difference to be mentioned is that the consequence of condition (7) clearly weakens the consequence of condition (1), and, symmetrically, the premise of condition (8) weakens the premise of condition (2). Obviously, (1) (resp., (8)) coincides with (1) (resp., (4)) for $\mu = \beta$ (cf. Example 3). If, additionally, $\mathcal{M}$ is a traditional integrity checking method that insists on the total integety premise, then both definitions coincide.

5.2 Cause-Based Integrity Checking

Obviously, Definition 4 does not indicate how $\mathcal{M}$ would compute its output. However, for each metric $(\mu, \preccurlyeq)$, condition 9, below, defines a μ-based method, as already proved in [5,7].

$$\mathcal{M}^{\mu}(D, IC, U) = sat \text{ iff } \mu(D^U, IC) \preccurlyeq \mu(D, IC). \tag{9}$$

Hence, for $\mu = \sigma$ or $\mu = \zeta$, we obtain two sound and complete cause-based inconsistency-tolerant integrity checking methods for controlling quality. That is illustrated by the following example. It also illustrates that different modes or degrees of inconsistency tolerance can be obtained by suitable choices of metrics.

Example 6. Let D and IC be as in Example 4. Further, suppose that *sheik* divorces from $wife_1$, and is about to wed with $wife_{n+1}$, as expressed by the update request $U = \{delete\ married(sheik, wife_1),\ insert\ married(sheik, wife_{n+1})\}$. Thus, $married(sheik, wife_{n+1}) \in D^U$ and $married(sheik, wife_{n+1}) \notin D$, hence $\sigma(D^U, IC) \nsubseteq \sigma(D, IC)$, hence $\mathcal{M}^{\sigma}(D, IC, U) = vio$. On the other hand, we clearly have $\zeta(D^U, IC) = \zeta(D, IC)$, hence $\mathcal{M}^{\zeta}(D, IC, U) = sat$.

6 Conclusion

We have revisited the idea to model the quality of stored data by integrity constraints. We have outlined how to measure the lack of data quality by focusing on causes, i.e., sets of data that are responsible for constraint violations.

In a previous approach that was also based on quality modeled by constraints [5,7], violated instances of constraints called cases had been measured. The basic idea of measuring quality by causes is very similar to the basic idea of measuring quality by cases: the less/more cases or causes of violated constraints exist, the better/worse is the quality of data. However, we have seen that cause-based metrics quantify the data that lack quality in a more directly and hence more accurately than the metrics based on cases. Moreover, we have shown that quality metrics, and in particular those based on causes, also serve to monitor and control the increase of unavoidable quality impairment.

Apart from our own previous work and the trivial inconsistency measure β, as characterized in Example 3, the ideas of quantifying the quality of databases by measuring the causes of integrity violation and of controlling the evolution of quality by inconsistency-tolerant measure-based integrity checking is original of this paper. Thus, the there is no further related work, except the basic paper on inconsistency measures in databases [8] and the literature on inconsistency tolerance in general, as discussed in [4,6].

The work presented in this paper essentially is of academical nature. A lot details remain open for making our theoretical ideas useful in practice. One example of many is given by the question of how to incorporate common practical constructs such as NULL values and aggregate functions into the concept of cause-based inconsistency-tolerant integrity checking. These and other issues are on the agenda for future investigations.

Other upcoming work of ours is going to deal with assigning application-specific weights to causes that violate cases of constraints. The purpose of that is to obtaining quality metrics that reflect the given application semantics in a more dedicated manner. Moreover, we are working on efficient ways to compute causes for the general case of databases and constraints that allow for negation in the body of clauses, as a basis for implementing cause-based inconsistency metrics. Preparatory theoretical studies in that direction have been initiated in [3].

References

1. Ceri, S., Gottlob, G., Tanca, L.: What you always wanted to know about Datalog (and never dared to ask). TKDE 1(1), 146–166 (1989)
2. Christiansen, H., Martinenghi, D.: On simplification of database integrity constraints. Fundam. Inform. 71(4), 371–417 (2006)
3. Decker, H.: Answers that Have Integrity in Databases that Violate Constraints. Presented at the ICALP Workshop SDKB 2010, to appear in the Post-Workshop Proceedings of SDKB (2011)

4. Decker, H.: Toward a uniform cause-based approach to inconsistency-tolerant database semantics. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2010. LNCS, vol. 6427, pp. 983–998. Springer, Heidelberg (2010)
5. Decker, H.: Quantifying the Quality of Stored Data by Measuring their Integrity. In: Proc. DIWT 2009, Workshop SMM, pp. 823–828. IEEE, Los Alamitos (2009)
6. Decker, H., Martinenghi, D.: Inconsistency-tolerant Integrity Checking. TKDE 23(2), 218–234 (2011)
7. Decker, H., Martinenghi, D.: Modeling, measuring and monitoring the quality of information. In: Heuser, C.A., Pernul, G. (eds.) ER 2009. LNCS, vol. 5833, pp. 212–221. Springer, Heidelberg (2009)
8. Grant, J., Hunter, A.: Measuring inconsistency in knowledgebases. J. Intelligent Information Systems 27(2), 159–184 (2006)
9. Ramakrishnan, R., Gehrke, J.: Database Management Systems. McGraw-Hill, New York (2003)

Complexity Measures in 4GL Environment

Csaba Nagy[1], László Vidács[2], Rudolf Ferenc[1], Tibor Gyimóthy[1], Ferenc Kocsis[3], and István Kovács[3]

[1] Department of Software Engineering, University of Szeged
[2] Research Group on Artificial Intelligence, University of Szeged & HAS
[3] SZEGED Software Zrt

Abstract. Nowadays, the most popular programming languages are so-called third generation languages, such as Java, C# and C++, but higher level languages are also widely used for application development. Our work was motivated by the need for a quality assurance solution for a fourth generation language (4GL) called Magic. We realized that these very high level languages lie outside the main scope of recent static analysis techniques and researches, even though there is an increasing need for solutions in 4GL environment.

During the development of our quality assurance framework we faced many challenges in adapting metrics from popular 3GLs and defining new ones in 4GL context. Here we present our results and experiments focusing on the complexity of a 4GL system. We found that popular 3GL metrics can be easily adapted based on syntactic structure of a language, however it requires more complex solutions to define complexity metrics that are closer to developers' opinion. The research was conducted in co-operation with a company where developers have been programming in Magic for more than a decade. As an outcome, the resulting metrics are used in a novel quality assurance framework based on the Columbus methodology.

Keywords: 4GL, Magic, software metrics, software complexity, software quality assurance.

1 Introduction

Programming languages are usually categorized into five levels or "generations" [1]. Solely binary numbers, the machine languages are the first generation languages (1GLs). Lower level programming languages (e.g. assembly) are the second generation languages (2GLs) and currently popular procedural and object-oriented languages are the third generation languages (3GLs). The higher level languages are all closer to human thinking and spoken languages. Using fourth generation languages (4GLs) a programmer does not need to write source code, but he can program his application at a higher level of abstraction, usually with the help of an application development environment. Finally, fifth generation languages (5GLs), would involve a computer which responds directly to spoken or written instructions, for instance English language commands.

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 293–309, 2011.

The main motivation of this work was to provide a quality assurance solution for a 4GL called Magic. Quality assurance tools are built heavily on software metrics, which reflect various properties of the analyzed system. Although several product metrics are already defined for mainstream programming languages, these metrics reflect the specialties of third generation programming languages. We faced the lack of software quality metrics defined for 4GLs. As we revealed the inner structure of Magic programs, we identified key points in defining new metrics and adapting some 3GL metrics to Magic. Our work was carried out together with a software company, where experts helped us in choosing the right definitions. The greatest challenge we faced was the definition of complexity metrics, where experienced developers found our first suggestions inappropriate and counterintuitive. Enhancing our measures we involved several developers in experiments to evaluate different approaches to complexity metrics.

In this paper we present our experiences in defining complexity metrics in 4GL environment, particularly in the application development environment called Magic, which was recently renamed to uniPaaS. Our contributions are:

– we adapted two most widespread 3GL complexity metrics to Magic 4GL (McCabe complexity, Halstead);
– we carried out experiments to evaluate our approaches (we found no significant correlation between developers ranking and our first adapted McCabe complexity, but we found strong correlation between a modified McCabe complexity, Halstead's complexity and between the developers ranking);
– as an outcome of the experiments we defined new, easily understandable and applicable complexity measures for Magic developers.

Supporting the relevance of the adapted metrics our experiment was designed to address the following research questions:

RQ1: Is there a significant correlation between adapted metrics of Magic programs?
RQ2: Is there a significant correlation between the complexity ranking given by developers and the ranking given by the adapted metrics?

The paper is organized as follows. First, in Section 2 we introduce the reader to the world of Magic and then in Section 3 we define our complexity metrics that were adapted to 4GL environment. Validating these metrics we carried out experiments which we describe in Section 4 and evaluate in Section 5. We discuss related work in Section 6 and finally we conclude in Section 7.

2 Specialties of 4GLs and the Magic Programming Language

It is important to understand the specialties of a fourth generation language before discussing its quality attributes. Hence, in this section we give an introduction into Magic as a fourth generation language. We present the basic structure of a typical Magic application and we discuss potential quality attributes of a Magic application.

Magic 4GL was introduced by Magic Software Enterprises (MSE) in the early 80's. It was an innovative technology to move from code generation to the use of an underlying meta model within an application generator.

2.1 The Structure of a Magic Application

Magic was invented to develop business applications for data manipulating and reporting, so it comes with many GUI screens and report editors. All the logic that is defined by the programmer, the layout of the screens, the pull down menus, reports, on-line help, security system, reside inside tables called Repositories. The most important elements of the meta model language are the various entity types of business logic, namely the Data Tables. A Table has its Columns and a number of Programs (consisting of subtasks) that manipulate it. The Programs or Tasks are linked to Forms, Menus, Help screens and they may also implement business logic using logic statements (e.g. for selecting variables, updating variables, conditional statements).

Focusing on the quality – especially on the complexity – of a Magic software, the most important language elements are those elements that directly implement the logic of the application. Figure 1 shows these most important language entities. A Magic *Application* consists of *Projects*, the largest entities dividing an application into separate logical modules. A Project has Data *Tables* and *Programs* (a top-level Task is called a Program) for implementing the main functionalities. A Program can be called by a *Menu* entry or by other Programs during the execution of the application. When the application starts up, a special program, the *Main Program* is executed. A *Task* is the basic unit for constructing a program. A Program can be constructed of a main task and subtasks in tree-structured task hierarchy. The Task represents the control layer of the application and its Forms represent the view layer. It typically iterates over a Table and this iteration cycle defines so-called *Logic Units*. For instance, a Task has a Prefix and a Suffix which represent the beginning and the ending of a Task, respectively. A record of the iteration is handled by the Record Main logic unit, and before or after its invocation the Record Prefix or Suffix is executed. A Logic Unit is the smallest unit which performs lower level operations (a series of *Logic Lines*) during the execution of the application. These operations can be simple operations, e.g. calling an other Task or Program, selecting a variable, updating a variable, input a data from a Form, output the data to a Form Entry.

Programming in Magic requires a special way of thinking. Basically, the whole concept is built on the manipulation of data tables which results in some special designs of the language. It can be seen that a Task belongs to an iteration over a data table so when a Task is executed it already represents a loop. Hence, the language was designed in a way that loops cannot be specified explicitly at statement level. It is also interesting that the expressions of a Task are handled separately so an expression can be reused more than once simply by referring to its identifier. For example, each Logic Line has a condition expression which

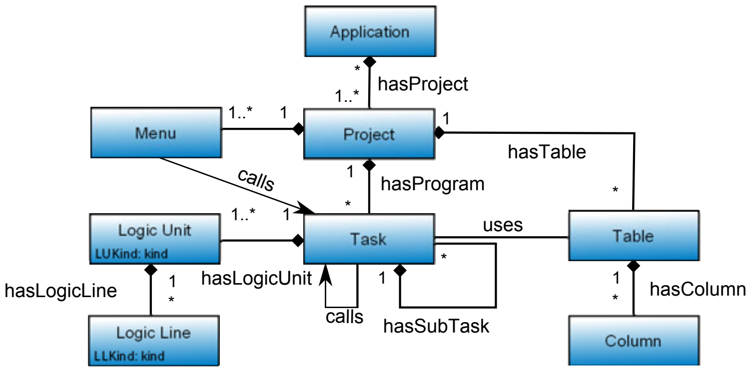


Fig. 1. Most important Magic schema entities

determines whether the operation should be executed or not. This condition can be easily maintained through the application development environment and the same expression may be easily used for more statements. So the developers are more comfortable in using conditional branches in the logic of an application. Consequently, they can easily see when the execution of statements belongs to the same condition even if the statements do not directly follow each other.

2.2 Measuring the Quality of a Magic Application

In previous projects [13], [14] we re-used and adapted elements of the Columbus methodology in the Magic environment. This methodology was successfully applied on object-oriented languages before [8] and today it covers the most influential areas of the software life cycle including the following goals [3]: decrease the number of post-release bugs, increase maintainability, decrease development/test efforts, assure sustainability though continuous measurement and assessment. Goals are targeted with continuous monitoring: scheduled analysis, data processing, storing and querying, visualization and evaluation. To accomplish these goals it is important to measure the characteristics of the software under question. For more details about Columbus methodology, please refer to our previous paper [3].

In case of third level languages, usually the best description of the software under question is its source code. It is obvious that the analysis of the source code is important to specify certain quality attributes. In case of fourth generation languages, developers do not necessarily write source code in the traditional way. In Magic, developers simply edit tables, use form editors, expression editors, etc. In such a language, the meta model of an application serves as a "source code" that can be analyzed for quality assurance purposes. Using this model we can describe the main characteristics of an application and we can locate potential coding problems or structures which may indicate bugs or bad design. We determined a number of product metrics for Magic and categorized them in *size*, *coupling*, and *complexity* groups. Most of them are based on popular and well-known product metrics such as the *Lines of Code*, *Number of Classes*, *Number of Attributes*, *Coupling Between Object classes* [4]. We realized that some metrics can be easily adapted from third generation languages, but their meaning and benefits for the developers may be completely different, compared to 3GL counterparts.

In case of size metrics, for instance, there is a possibility to identify a series of "Number of" metrics (e.g. *Number of Programs*, *Menus*, *Helps*), but they are considered less useful and interesting for the developers. The reason for that is that these numbers can be easily queried through the application development environment. The Lines of Code (*LOC*) metric can be easily adapted by taking into account that the *Logical Line* language entity of Magic can be corresponded to a "Line of Code" in a third generation language. However, the adapted metric should be used with caution because it carries a different meaning compared to the original *LOC* metric. In 3GLs *LOC* typically measures the size of the whole system and it is used to estimate the programming effort in different effort models (e.g. *COCOMO* [5]). In case of Magic, a project is built on many repositories (Menus, Help screens, Data Tables, etc.) and *LOC* measures just one size attribute of the software (the Program repository). Hence, *LOC* is not the sole size attribute of an application so it cannot be used alone for estimating the total size of the full system. It is interesting to note that when 4GLs became popular, many studies were published in favor of their use. These studies tried to predict the size of a 4GL project and its development effort, for instance by calculating function points [16],[17] or by combining 4GL metrics with metrics for database systems [10].

Coupling is also interesting in a 4GL environment. In object-oriented languages a typical metric for coupling is the *Coupling Between Object classes* (*CBO*) metric which provides the number of classes to which a given class is coupled. A class is coupled to another one if it uses its member functions and/or instance variables. 4GLs usually do not have language elements representing objects and classes. For instance in Magic, there are no entities to encapsulate data and related functionalities, however there are separated data entities (Tables) and their related functionalities are specified in certain Tasks or Programs. Therefore it makes sense measuring the *Coupling Between Tasks and Data Tables*, not unlike the *Coupling Between Tasks and Tasks.*

3 Measuring the Complexity of Magic Applications

We identified different quality attributes and defined a bunch of metrics for Magic applications. Simple size and coupling metrics reflected well the opinion of the developers, but this was not the case for complexity metrics. It was our biggest challenge to measure the complexity of a 4GL system. There are many different approaches for third generation languages [6]. At source code level, well known approaches were developed by McCabe [11] and Halstead [9], which are widely used by software engineers, e.g., for software quality measurement purposes and for testing purposes.

We adapted McCabe's cyclomatic complexity and Halstead's complexity metrics in 4GL environment, but when we showed the results to developers, their feedback was that all the programs that we identified as most complex programs in their system are not that much complex according to their experience. We note here that all the programmers have been programming in Magic for more than 3 years (some of them for more than a decade) and most of them were well aware of the definition of structural complexity [1], but none of them have heard before about cyclomatic or Halstead complexity.

3.1 McCabe's Cyclomatic Complexity Metric

In this section we present our adaptations of complexity metrics and a modified cyclomatic complexity measure.

First, we adapted McCabe's complexity metric [11] to Magic. McCabe used a graph-theory measure, the *cyclomatic number* to measure the complexity of the control flow of a program. It was shown that of any structured program with only one entrance and one exit point, the value of McCabe's cyclomatic complexity is equal to the number of decision points (i.e., the number of "if" statements and conditional loops) contained in that program plus one.

McCabe's complexity is usually measured on method or function level. For object-oriented languages it is possible to aggregate complexities of methods to class level. The idea of *Weighted Methods per Class* (*WMC*) [7] is to give weights to the methods and sum up the weighted values. As a complexity measure this metric is the sum of cyclomatic complexities of methods defined in a class. Therefore *WMC* represents the complexity of a class as a whole.

In case of Magic, the basic operations are executed at Logic Unit level. A Logic Unit has its well-defined entry and exit point too. Likewise, a Task has predefined Logic Units. That is, a Task has a Task Prefix, Task Suffix, Record Prefix, Record Main, Record Suffix, etc. This structure is similar to the construction of a Class where a Class has some predefined methods, e.g., constructors and destructors. Hence, we defined McCabe's complexity at Logic Unit level with the same definition as it is defined for methods (see definition of $McCC(LU)$ in Definition 1). So it can be simply calculated by counting the statements with preconditions (i.e., the branches in the control flow) in a Logic Unit. Likewise, the complexity of a Task can be measured by summing up the complexity values of its Logic Units. We call this complexity measure as the *Weighted Logic Units per Task* (see $WLUT(T)$ in Definition 2).

$McCC(LU) = Number\ of\ decision\ points\ in\ LU + 1.$
LU: a Logic Unit of a Task

Def. 1: The definition of McCabe's cyclomatic complexity for Logic Units

The $McCC(LU)$ and $WLUT(T)$ metrics were adapted directly from the 3GL definitions simply based on the syntactic structure of the language. When we

$WLUT(T) = \sum_{LU \in T} McCC(LU)$
T: a Task in the Project
LU: a Logic Unit of T

Def. 2: The definition of Weighted Logic Units per Task (WLUT)

first showed the definitions to the developers they agreed with them and they were interested in the complexity measures of their system. However, the results did not convince them. Those Tasks that we identified as the most complex tasks of their system were not complex according to the developers, not unlike, those tasks that were identified complex by the developers had lower $WLUT$ values.

Developers suggested us, that in addition to the syntactic structure of the language, we should add the semantic information that a Task is basically a loop which iterates over a table and when it calls a subtask it is rather similar to an embedded loop. This semantic information makes a Task completely different from a Class. Considering their suggestion we modified the McCabe complexity as follows ($McCC_2$). For a Logic Unit we simply count the number decision points, but when we find a call for a subtask it is handled as a loop and it increases the complexity of the Logic Unit by the complexity of the called subtask. That is, the complexity of a Task is the sum of the complexity of its Logic Units. For the formalized definition see Definition 3.

$McCC_2(LU) = \mathit{Number\ of\ decision\ points\ in\ LU} + \sum_{TC \in LU} McCC_2(TC) + 1.$
$McCC_2(T) = \sum_{LU \in T} McCC_2(LU)$
LU: a Task of the Project
LU: a Logic Unit of T
TC: a called Task in LU

Def. 3: The definition of the modified McCabe's cyclomatic complexity ($McCC_2$)

The main difference between $WLUT(T)$ and $McCC_2(T)$ is that $McCC_2(T)$ takes into account the complexity of the called subtasks too in a recursive way. A recursive complexity measure would be similar for procedural languages when a function call would increase the complexity of the callee function by the complexity of the called function. (Loops in the call graph should be handled.)

Developers found the idea of the new metric more intuitive as it takes into account the semantics too. Later, in our experiments we found that the new metric correlates well with the complexity ranking of the developers (see Section 4).

3.2 Halstead's Complexity Metrics

Some of the developers also complained that our metrics do not reflect the complexity of the expressions in their programs. It should be noted here that Magic handles the expressions of a Task separately. An expression has a unique identifier and can be used many times inside different statements simply by referring to its identifier. The application development environment has an expression editor for editing and handling expressions separately. This results in a coding style where developers pay more attention on the expressions they use. They see the list of their expressions and large, complex ones may be easily spotted out.

Halstead's complexity metrics [9] measure the complexity of a program based on the lexical counts of symbols used. The base idea is that complexity is affected by the used operators and their operands. Halstead defines four base values for measuring the number of distinct and total operands and operators in a program (see Definition 4). The base values are constituents of higher level metrics, namely, *Program Length* (HPL), *Vocabulary size* (HV), *Program Volume* (HPV), *Difficulty level* (HD), *Effort to implement* (HE). For the formalized definitions see Definition 5.

n_1: the number of distinct operators
n_2: the number of distinct operands
N_1: the total number of operators
N_2: the total number of operands

Def. 4: Base values for measuring the number of distinct and total operands and operators in a program

$$HPL = N_1 + N_2$$
$$HV = n_1 + n_2$$
$$HPV = HPL * log_2(HV)$$
$$HD = (\frac{n_1}{2}) * (\frac{N_2}{n_2})$$
$$HE = HV * HD$$

Def. 5: Halstead's complexity measures

In case of Magic, symbols may appear inside expressions so the choice of Halstead's metrics seemed appropriate for measuring the complexity of expressions. Operands can be interpreted as the symbols like in a 3GL language (e.g. variable names, task identifiers, table identifiers) and operators are the operators (plus, minus, etc.) inside expressions.

Later, in our experiments we found that the Halstead's metrics correlate with the complexity ranking of the developers (see Section 4), but the modified McCabe's complexity is closer to the opinion of the developers.

4 Experiments with Complexity Metrics

Although the classic complexity metrics are successfully adapted to the Magic language, there are no empirical data available on how they relate to each other and on their applicability in software development processes. We observed that, except the McCabe metric, complexity metrics generally do not have a justified conceptual foundation. Rather, they are defined based on experience [18]. We plan to fill in the gap first, by calculating and evaluating the adapted metrics on industrial size programs to see their relations; second, by surveying experts at a Magic developer company to see the usability of the definitions. We emphasize the importance of feedback given by Magic experts. There is no extensive research literature on the quality of Magic programs. Hence, the knowledge accumulated during many years of development is essential to justify our metrics.

Thus, to evaluate our metrics, metrical values were computed on a large-scale Magic application, and a questionnaire was prepared for experienced Magic developers to see their thoughts on complexity. We sought for answers for the following research questions:

RQ1: *Is there a significant correlation between adapted metrics of Magic programs?*
RQ2: *Is there a significant correlation between the complexity ranking given by developers and the ranking given by the adapted metrics?*

We performed static analysis and computed metrics on a large-scale application using the ***MAGISTER*** system [13] (see Table 1). There are more than 2,700 programs in the whole application, which is a huge number in the world of Magic. The total number of non-Remark Logic Lines of this application is more than 300,000. The application uses more than 700 tables.

Table 1. Main characteristics of the system under question

Metric	Value
Number of Programs	2 761
Number of non-Remark Logic Lines	305 064
Total Number of Tasks	14 501
Total Number of Data Tables	786

There were 7 volunteer developers taking part in the survey at the software developer company. The questionnaire consisted of the following parts:

1. Expertise:
   (a) Current role in development.
   (b) Developer experience in years.
2. Complexity in Magic:
   (a) At which level of program elements should the complexity be measured?
   (b) How important are the following properties in determining the complexity of Magic applications? (List of properties is given.)
   (c) Which additional attributes affect the complexity?
3. Complexity of concrete Magic programs developed by the company.
   (a) Rank the following 10 Magic programs (most complex ones first).

The most important part of the questionnaire is the ranking of the concrete programs. This makes possible comparing what is in the developers' mind to the computed metrics. Subject programs for ranking were selected by an expert of the application. He was asked to select a set of programs which *a)* is representative to the whole application, *b)* contains programs of various size, *c)* developers are familiar with. He was not aware of the purpose of selection. The selected programs and their main size measures can be seen in Table 2. The number of programs is small as we expected a solid, established opinion of participants in a reasonable time. In the table the Total Number of Logic Lines (containing task hierarchy) (*TNLL*), the Total Number of Tasks (*TNT*), Weighted Logic Units per Task (*WLUT*) and the cyclomatic complexity ($McCC_2$) are shown.

Table 2. Selected programs with their size and complexity values

Id	Name	*TNLL*	*TNT*	*WLUT*	$McCC_2$
69	Engedmény számítás egy tétel	1352	24	10	214
128	TESZT:Engedmény/rabatt/formany	701	16	14	63
278	TÖRZS:Vevő karbantartó	3701	129	47	338
281	TÖRZS:Árutörzs összes adata	3386	91	564	616
291	Ügyfél zoom	930	29	8	27
372	FOK:Fökönyv	1036	31	113	203
377	Előleg bekérő levél képzése	335	6	5	20
449	HALMOZO:Havi forgalom	900	22	3	117
452	HALMOZO:Karton rend/vissz	304	9	4	34
2469	Export_New	7867	380	382	761

5 Results

We first discuss our findings about complexity measurements gathered via static analysis of the whole application. Later, we narrow down the set of observed programs to those taking part in the questionnaire, and finally we compare them to the opinion of the developers.

5.1 RQ1: *Is there a significant correlation between adapted metrics of Magic programs?*

Here we investigate the correlation between the previously defined metrics.The McCabe and Halstead metrics are basically different approaches, so first we investigate them separately.

Halstead metrics. Within the group of Halstead metrics significant correlation is expected, because – by definition – they depend on the same base measures. In spite of that, different Halstead measures capture different aspects of computational complexity. We performed a Pearson correlation test to see their relation in Magic. Correlation values are shown in Table 3. Among the high expected

Table 3. Pearson correlation coefficients (R^2) of Halstead metrics and the Total Number of Expressions (*TNE*) (all correlations are significant at 0.01 level)

	HPL	*HPV*	*HV*	*HD*	*HE*	*TNE*
HPL	1.000	0.906	0.990	0.642	0.861	0.769
HPV	0.906	1.000	0.869	0.733	0.663	0.733
HV	0.990	0.869	1.000	0.561	0.914	0.773
HD	0.642	0.733	0.561	1.000	0.389	0.442
HE	0.861	0.663	0.914	0.389	1.000	0.661

Table 4. Pearson correlation coefficients (R^2) of various complexity metrics (all correlations are significant at 0.01 level)

	WLUT	$McCC_2$	*HPV*	*NLL*	*TNT*
WLUT	1.000	0.007	0.208	0.676	0.166
$McCC_2$	0.007	1.000	0.065	0.020	0.028
HPV	0.208	0.065	1.000	0.393	0.213

correlation values, *HD* and *HE* metrics correlate slightly lower with the other metrics. We justified Halstead metrics using the *Total Number of Expressions* (*TNE*), which can be computed in a natural way as expressions are separately identified language elements. The relatively high correlation between *TNE* and other Halstead metrics shows that the *TNE* metric is a further candidate for a complexity metric. This reflects suggestions of the developers too. For the sake of simplicity, in the rest of this paper we use the *HPV* metric to represent all five metrics of the group.

Comparison of adapted complexity metrics. Table 4 contains correlation data on McCabe-based complexity (*WLUT*, $McCC_2$), *HPV* and two size metrics. The three complexity measures has significant, but only a slight correlation, which indicates that they show different aspects of the program complexity.

We already presented the differences between *WLUT* and $McCC_2$ before. The similar definitions imply high correlation between them. Surprisingly, based on the measured 2700 programs their correlation is the weakest (0.007) compared to other metrics so they are almost independent. $McCC_2$ is measured on the subtasks too, which in fact affects the results. Our expectation was that, for this reason, $McCC_2$ has a stronger correlation with *TNT* than *WLUT*. However, the $McCC_2$ metric only slightly correlates with *TNT*. This confirms that developers use many conditional statements inside one task, and the number of conditional branches has a higher impact on the $McCC_2$ value.

Rank-based correlation. From this point on, we analyze the rank-based correlation of metrics. The aim is to facilitate the comparison of results to the ranks given by the developers. The number of considered programs is now narrowed down to the 10 programs mentioned before in Section 4. Ranking given by a certain metric is obtained in the following way: metric values for the 10 programs are computed, programs with higher metric values are ranked lower (e.g. the program with highest metric value has a rank no. 1). The selection of 10 programs is justified by the fact, that the previously mentioned properties (e.g. different sizes, characteristics) can be observed here as well. In Figure 2, the ranking of Halstead metrics is presented. On the x axis the programs are shown (program Id), while their ranking value is shown on the y axis (1-10). Each line represents a separate metric. Strong correlation can be observed as the values are close to each other. Furthermore, the HD and HE metrics can also be visually identified as a little bit outliers. (Note: Spearman's rank correlation values are also computed.) The ranking determined by the three main complexity metrics can be seen in Figure 3. The x axis is ordered by the $McCC_2$ complexity, so programs with lower $McCC_2$ rank (and higher complexity) are on the left side. The similar trend of the three metrics can be observed, but they behave in a controversial way locally.

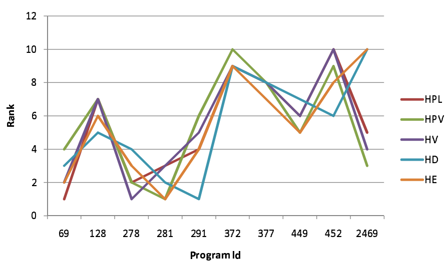


Fig. 2. Ranking of Halstead complexity metrics (ordered by program ID)

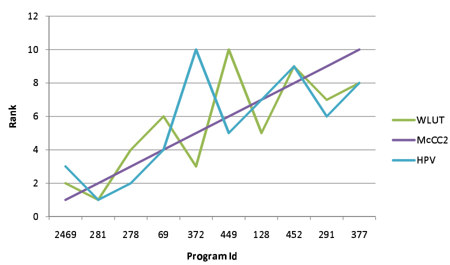


Fig. 3. Ranking of main complexity metrics (ordered by $McCC_2$)

Answering our research question, we found that some of the investigated complexity measures are in strong correlation, but some of them are independent measures. We found strong correlation between the Halstead metrics and we also found that these metrics correlate to the Total Number of Expressions. We found that our first adaptation of cyclomatic complexity ($WLUT$) has only a very weak correlation to our new version ($McCC_2$), which correlates well with other measures. This also confirms that the new measure might be a better representation of the developers opinion about complexity.

5.2 RQ2: *Is there a significant correlation between the complexity ranking given by developers and the ranking given by the adapted metrics?*

In the third part of the questionnaire developers were asked to give an order of the 10 programs which represents their complexity order. Previously, developers were given a short hint on common complexity measures, but they were asked to

Table 5. Correlation of Magic complexity metrics and developers' view (Spearman's ρ^2 correlation coefficients, marked values are significant at the 0.05 level)

	$WLUT$	$McCC_2$	HPV	HE	EC
$WLUT$	1.000	0.575	0.218	0.004	0.133
$McCC_2$	0.575	1.000	0.520	0.027	0.203
HPV	0.218	0.520	1.000	0.389	0.166
HE	0.004	0.027	0.389	1.000	0.497
EC	0.133	0.203	0.166	0.497	1.000

express their subjective opinion too. Most of the selected programs were probably familiar to the developers since the application is developed by their company. Furthermore they could check the programs using the development environment during the ranking process.

Ranks given by the 7 developers are shown in Figure 4, where each line represents the opinion of one person. It can be seen that developers set up different ranks. There are diverse ranks especially in the middle of the ranking, while the top 3 complex programs are similarly selected. Accordingly, developers agree in the least complex program, which is 2469. Correlations of developers' ranks were also computed. Significant correlation is rare among the developers, only ranks of P4, P5 and P6 are similar (Pi denotes a programmer in Figure 4).

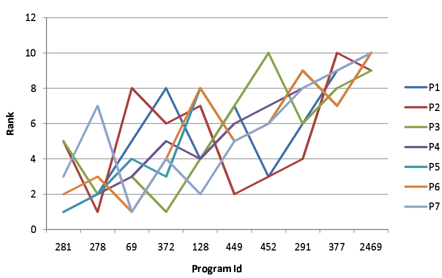


Fig. 4. Ranks given by Magic experts

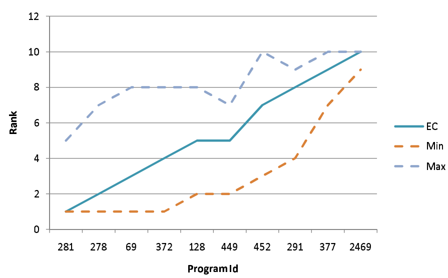


Fig. 5. The EC value, min and max ranks

We defined the EC value (*Experiment Complexity*) for each selected program as the rank based on the average rank given by developers. In Figure 5 the EC value is shown together with min and max ranks of the developers. We note that summarizing the developers' opinion in one metric may result in loosing information since developers may had different aspects in their minds. We elaborate on this later in the Threats to Validity section. We treat this value as the opinion of the developer community.

We compared the EC value to the previously defined complexity metrics. Table 5 contains correlation values of main metrics. The EC value shows significant correlation only with the HE measure.

Besides statistical information, complexity ranks are visualized as well. We found that the rank based correlation obscures an interesting relation between $McCC_2$ and the EC value. Ranks for each program are shown in Figure 6. The order of programs follows the $McCC_2$ metric. Despite that Spearman's ρ^2 values show no significant correlation, it can be clearly seen that developers and $McCC_2$ metric gives the same ranking, except for program 2469. This program is judged in an opposite way. The program contains many decision points, however developers say that it is not complex since its logic is easy to understand. According to the HE metric, this program is also ranked as the least complex.

Answering our research question we found that the rankings given by adapted metrics have significant and sometimes surprisingly strong relation to the ranking given by developers, except for the $WLUT$ metric. Halstead's metrics have a significant correlation here, especially the HE metric. However, the strongest relation was discovered in case of the $McCC_2$ metric.

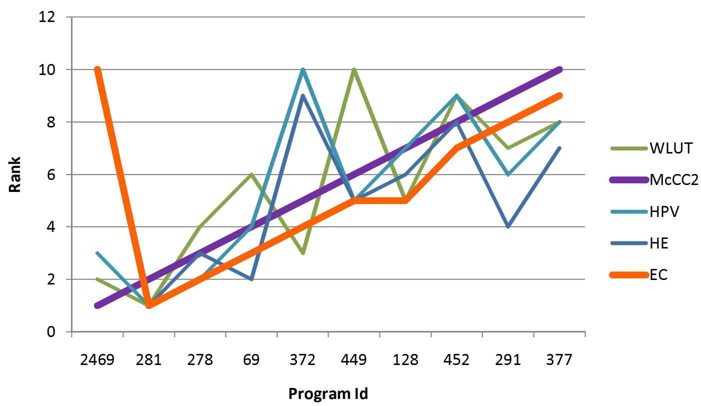


Fig. 6. The EC value compared to the main complexity metrics

5.3 Discussion of the Limitations

Although we carefully designed our experiments, there are some points which may affect our results and observations. Complexity metrics were computed on a large-scale and data-intensive application, but the results may be affected by coding style and conventions of a single company. Measurements of Magic applications from other domains and developer companies are needed. This applies to the questionnaire as well. The number of participants and selected programs should be increased to draw general conclusions. Programs were selected by a person, not randomly based on a specific distribution, which may also affect our results. Evaluation of developers' view is done by means of ranking, which results in loss of information in transforming measured values into ranks. The EC value is an average rank given by the developers. It would be more realistic to formalize their viewpoints during the ranking process.

6 Related Work

We cited related papers before when we elaborated on our metrics and experiments. We note here, that there are many different approaches for measuring the complexity of a software at source code level. First, and still popular complexity measures (McCabe [11], Halstead [9], Lines of Code [2]) was surveyed by Navlakha [15]. A recent survey which sums up todays complexity measures was published by Sheng Yu et al. [18]. In 4GL environment, to our best knowledge, there were no previous researches to measure structural complexity attributes of a Magic application. Even though, for other 4GLs there are some attempts to define metrics to measure the size of a project [16], [17], [10]. There are also some industrial solutions to measure metrics in 4GL environment. For instance *RainCode Roadmap*[1] for Informix 4GL provides a set of predefined metrics about code complexity (number of statements, cyclomatic complexity, nesting level), about SQLs (number of SQL statements, SQL tables, etc.), and about lines (number of blank lines, code lines, etc.). In the world of Magic, there is a tool for optimization purposes too called Magic Optimizer[2] which can be used to perform static analysis of Magic applications. It does not measure metrics, but it is able to locate potential coding problems which also relates to software quality.

In 3GL context there are also papers available to analyze the correlation between certain complexity metrics. For instance, Meulen et al. analyzed about 71,917 programs from 59 fields written in C/C++ [12]. Their result showed that there are very strong connections between LOC and HCM, LOC and CCM. Our work found also similar results, but our research was performed in a 4GL context with newly adapted complexity metrics. We additionally show, that in our context traditional metrics have totally different meanings for the developers.

7 Conclusions and Future Work

The main scope of our paper was to adapt most widespread 3GL structural complexity metrics (McCabe's cyclomatic complexity and Halstead's complexity measures) to a popular 4GL environment, the Magic language. We introduced the specialties of Magic and we presented formal definitions of our metrics in 4GL environment. Besides the simple adaptation of the metrics, we presented a modified version of McCabe's cyclomatic complexity ($McCC_2$), which measured the complexity of a task by aggregating the complexity values of its called subtasks too. We addressed research questions about our new metrics whether they are in relation with developers' complexity ranking or not. We designed and carried out an experiment to answer our questions and we found that:

- there is significant correlation among all the investigated metrics, and there is strong correlation between the Halstead measures which also correlate to the Total Number of Expressions;

[1] `http://www.raincode.com/fglroadmap.html`

[2] `http://www.magic-optimizer.com/`

– the rankings given by adapted metrics have significant and very strong relation to the ranking given by developers (especially in case of the $McCC_2$, but except for the $WLUT$ metric).

As an outcome, we found also that our modified measure has a strong correlation with developers' ranking.

To sum up the conclusions of our work, we make the following remarks:

– We made advancement in a research area where no established metrics (previous similar measurements and experience reports) were available.
– We successfully adapted 3GL metrics in a popular 4GL environment, in the Magic language.
– We evaluated our metrics by the developers in a designed experiment and metrics were found easily understandable and useful.
– A modified version of the McCabe's cyclometic complexity was found to reflect surprisingly well the ranking given by the developer community.

Besides gathering all the previously mentioned experiences, the defined metrics are implemented as part of a software quality assurance framework, namely the ***MAGISTER***[3] system which was designed to support the development processes of an industrial Magic application.

About our future plans, as we offer quality assurance services, we expect to gain data from other application domains to extend our investigations. Most importantly we plan to set up appropriate baselines for our new metrics in order to better incorporate them into the quality monitoring process of the company and into the daily use.

Acknowledgements

We thank István Siket, Dániel Fritsi, Ferenc Smohai and the volunteer developers for their contributions. This research was supported by the Hungarian national grant GOP-1.1.2-07/1-2008-0007, TECH 08-A2/2-2008-0089, GOP-1.1.1-07/1-2008-0081, OTKA K-73688 and by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

References

1. IEEE Standard Glossary of Software Engineering Terminology. Tech. rep. (1990)
2. Albrecht, A.J., Gaffney, J.E.: Software function, source lines of code, and development effort prediction: A software science validation. IEEE Transaction on Software Engineering 9, 639–648 (1983)
3. Bakota, T., Beszédes, Á., Ferenc, R., Gyimóthy, T.: Continuous software quality supervision using SourceInventory and Columbus. In: ICSE Companion, pp. 931–932 (2008)

[3] `http://www.szegedsw.hu/magister`

4. Basili, V.R., Briand, L.C., Melo, W.L.: A validation of object-oriented design metrics as quality indicators. IEEE Transaction on Software Engineering 22, 751–761 (1996)
5. Boehm, B.W.: Software Engineering Economics, 1st edn. Prentice Hall PTR, Upper Saddle River (1981)
6. Burgin, M., Debnath, N.: Complexity measures for software engineering. J. Comp. Methods in Sci. and Eng. 5, 127–143 (2005)
7. Chidamber, S.R., Kemerer, C.F.: A metrics suite for object oriented design. IEEE Transaction on Software Engineering 20, 476–493 (1994)
8. Ferenc, R., Beszédes, Á., Tarkiainen, M., Gyimóthy, T.: Columbus – Reverse Engineering Tool and Schema for C++. In: Proceedings of the 18th International Conference on Software Maintenance (ICSM 2002), pp. 172–181. IEEE Computer Society, Los Alamitos (2002)
9. Halstead, M.H.: Elements of Software Science (Operating and programming systems series). Elsevier Science Inc., New York (1977)
10. MacDonell, S.: Metrics for database systems: An empirical study. In: Proceedings of the 4th International Symposium on Software Metrics, pp. 99–107. IEEE Computer Society, Los Alamitos (1997)
11. McCabe, T.J.: A complexity measure. IEEE Transaction on Software Engineering SE-2(4) (December 1976)
12. van der Meulen, M.J.P., Revilla, M.A.: Correlations between internal software metrics and software dependability in a large population of small C/C++ programs. In: Proceedings of ISSRE 2007, The 18th IEEE International Symposium on Software Reliability, pp. 203–208 (November 2007)
13. Nagy, C.: MAGISTER: Quality assurance of magic applications for software developers and end users. In: Proceedings of ICSM 2010, 26th IEEE International Conference on Software Maintenance, pp. 1–6. IEEE Computer Society, Los Alamitos (2010)
14. Nagy, C.: Solutions for reverse engineering 4GL applications, recovering the design of a logistical wholesale system. In: Proceedings of CSMR 2011, 15th European Conference on Software Maintenance and Reengineering. IEEE Computer Society, Los Alamitos (2011)
15. Navlakha, J.K.: A survey of system complexity metrics. The Computer Journal 30, 233–238 (1987)
16. Verner, J., Tate, G.: Estimating size and effort in fourth-generation development. IEEE Software 5, 15–22 (1988)
17. Witting, G.E., Finnie, G.R.: Using artificial neural networks and function points to estimate 4GL software development effort. Australasian Journal of Information Systems 1(2) (1994)
18. Yu, S., Zhou, S.: A survey on metric of software complexity. In: Proceedings of ICIME 2010, The 2nd IEEE International Conference on Information Management and Engineering, pp. 352–356 (April 2010)

A Framework for Integrated Software Quality Prediction Using Bayesian Nets

Łukasz Radliński

Department of Information Systems Engineering, University of Szczecin
ul. Mickiewicza 64, 71-101 Szczecin, Poland
lukrad@uoo.univ.szczecin.pl

Abstract. The aim of this study is to develop a framework for integrated software quality prediction. This integration is reflected by a range of quality attributes incorporated in the model as well as relationships between these attributes. The model is formulated as a Bayesian net, a technique that has already been used in various software engineering studies. The framework enables to incorporate expert knowledge about the domain as well as related empirical data and encode them in the Bayesian net model. Such model may be used in decision support for software analysts and managers.

Keywords: Bayesian net, framework, quality model, quality features, software quality prediction.

1 Introduction

Software plays an ever-increasing role for many people living in civilized world. This causes the demand for delivering software of high quality. Reaching this target depends on various conditions, among which the delivery of information for decision support is one of the fundamentals. Most often such information is provided using the models for assessment or prediction of software quality.

Software quality has been investigated since the turn of 1960's and 1970's [3], [33], [34]. Extensive research and industrial work led to significant results in software quality analysis and modeling. However, many proposed models for software quality prediction have been built for specific environments and may not be easily adapted. On the other hand, it is possible to build a general-purpose model. However, there is a danger that it will lack the necessary level of details.

Hence, the main aim of this paper is to develop a framework for building predictive models for software quality. With this framework is it possible to build customized models that may be used as decision support tools for software analysts and managers. To enhance the explanatory potential, these models need to incorporate a range of variables and relationships between them, including various levels of quality features, development process factors, and project factors.

The motivation for developing this framework is the lack of predictive models for integrated software quality prediction with such high analytical potential. Software

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 310–325, 2011.

quality models covering a variety of quality features [11], [13], [20] typically have low analytical potential for decision support. They are only frames for developing analytical or predictive models. Conversely, models with high analytical potential typically focus on a single or limited quality features, for example on reliability [22], [24]. Proposed framework integrates both approaches and enables building customized models for integrated quality prediction and decision support.

We have selected Bayesian net (BN) as the underlying formalism for a predictive model. BNs are “well-suited to capturing vague and uncertain knowledge” [27] common for software quality area. Section 3.1 discusses detailed advantages of BNs.

This paper makes the following main contributions:

1. A framework for developing BN models for integrated software quality prediction. This framework may also be used in developing BN models for similar problems in of decision making in areas other than software project management.
2. A Bayesian Net for Integrated Software Quality prediction (BaNISoQ). We built this model using proposed framework and then performed internal validation.

The framework proposed here does not directly implement any software quality model. However, the BaNISoQ model, that is a proof of concept for this framework, refers to the set of ISO standards.

The rest of this paper is organized as follows: Section 2 brings closer the area of software quality by providing its definition and the hierarchy of quality attributes. Section 3 introduces Bayesian nets by discussing the main motivations for selecting this technique, providing its formal definition and discussing existing related models. Section 4 illustrates proposed framework for building a model for integrated software quality prediction. Section 5 focuses on presenting such predictive model by illustrating its structure and parameter definition. Section 6 focuses on model validation by explaining model behavior in three simulation analyses. Section 7 discusses how proposed framework may be used in other areas or fields. Section 8 considers the limitations and threats to validity of proposed approach. Section 9 draws the conclusions and discusses future work.

2 Software Quality

There are various definitions of software quality, which reflect different points of view and highlight diverse aspects of quality. Typically, they refer to user needs, expectations or requirements [11], [19], [20], [25], [35]. This study follows the ISO definition of software quality as a “degree to which the software product satisfies stated and implied needs when used under specified conditions” [11].

The authors of various studies on software quality developed models for software quality assessment. All of the widely accepted models treat software quality not as a single value but rather as a set or detailed hierarchy of quality features, sub-features and measures. Typically, these standards and quality models use different terminology for calling the elements of this hierarchy, such as characteristics, attributes, factors, constructs, dimensions – for the main-level elements. In this study ‘features’ refer to the main-level elements, ‘sub-features’ are elements of ‘features’, and 'measures' are elements of 'sub-features' that often can be directly measured. Other main differences between these models of software quality include:

- different sets of features, sub-features and measures,
- different definitions of features, sub-features and measures,
- different relationships between features, sub-features and measures.

This study strongly refers to the set of ISO standards for software quality. Specifically, these include the family of 9126 standards [13], [14], [15], [16], revised by the new family of 250xx standards, mainly [11], [12], some of which are still under development. We have selected these standards mainly because they are the newest, thus refer to 'modern' software, and are maintained by standardizing organization (ISO), thus are popular and widely accepted. Detailed discussion on selecting quality models may be found in [6]. Proposed framework may use other models of software quality, including custom models, without the loss of details.

For the purpose of this study, software quality is expressed as a set of the following features and their sub-features:

- functional suitability – functional appropriateness, accuracy, functional suitability compliance,
- reliability – maturity, availability, fault tolerance, recoverability, reliability compliance,
- security – confidentiality, integrity, non-repudiation, accountability, security compliance,
- compatibility – co-existence, interoperability, compatibility compliance,
- operability – appropriateness recognisability, learnability, ease of use, attractiveness, technical accessibility, operability compliance,
- performance efficiency – time behavior, resource utilization, performance efficiency compliance,
- maintainability – modularity, reusability, analyzability, changeability, modification stability, testability, maintainability compliance,
- portability – adaptability, installability, replaceability, portability compliance,
- usability – effectiveness, efficiency, satisfaction, usability compliance,
- flexibility – context conformity, context extendibility, accessibility, flexibility compliance,
- safety – operator health and safety, commercial damage, public health and safety, environmental harm, safety compliance.

The last three features refer to quality in use while the remaining to the internal (i.e. not relying on software execution) and external (i.e. relevant to running software) quality metrics. Due to the publication limits, we cannot provide detailed definitions of sub-features. More details on software quality, including detailed definitions, assessment and management issues, may be found in [17], [19], [20], [25], [35].

These software quality models are typically used as bases in the *assessment* of software quality based on various measures of software product. However, it is possible to convert these models to *predictive* models by defining relationships numerically and adding appropriate measures, as shown in this study. These measures should not only describe software product but also the process of software development, for example effort allocation, process and people quality, tool usage.

3 Bayesian Nets

3.1 Motivations for Bayesian Nets

Various modeling techniques have been used in empirical software engineering. These include multiple regression, neural networks, rule-based models, decision trees, system dynamics, estimation by analogy, Bayesian analysis, support vector machines. Earlier studies [23], [29], [39] discuss and compare these techniques in detail.

The most important feature of the majority of these techniques is that they are data-driven. It means that, to generate a predictive model or use specific 'predictive' algorithm, they require a dataset with empirical data. While such techniques have numerous advantages, we could not use them for this study. The primary reason is that in an extensive search for publicly available empirical datasets with a variety of quality features, we have not found any useful for this study. Cooperation with various software companies revealed that software companies typically do not gather such data of required volume and granularity. Thus, building a model for integrated software quality prediction requires incorporating expert knowledge.

Understanding the difficulty of the approach, this study follows the use of Bayesian nets (BNs) as a modeling technique. Selecting BNs have been justified by the unique set of advantages that they share. First, a model may incorporate a mixture of expert knowledge and empirical data. Second, a model may contain causal relationships between variables (typically defined by experts). Third, variables are defined probabilistically to enable explicit incorporation of uncertainty about the modeled world. Fourth, there is no fixed and explicit distinction of independent and dependent variables – in a particular scenario a variable with assigned observation becomes an independent variable, which is used to predict the states of all variables without observations (i.e. dependent). This feature is related with the ability to run a model with incomplete data and the ability of both forward and backward inference. Finally, BNs are graphical models so the relationships may be visualized to improve understandability and clearness for end-user.

3.2 Formal Definition of Bayesian Net

A Bayesian net is a model, which consists of the following:

- a directed acyclic graph with nodes $\mathbf{V} = \{v_1, \ldots, v_n\}$ and directed links $\mathbf{E}$,
- a set of random variables $\mathbf{X}$, represented by the nodes in graph,
- a set of conditional probability distributions $\mathbf{P}$, containing one distribution $P(X_v|X_{pa(v)})$, for each random variable $X_v \in \mathbf{X}$,

where $pa(v)$ denotes a set of parents for a node v. Bayesian net encodes a joint probability distribution $P(\mathbf{X})$ over a set of random variables $\mathbf{X}$. The set of conditional probability distributions $\mathbf{P}$ specifies a multiplicative factorization of the joint probability distribution over $\mathbf{X}$: $P(\mathbf{X}) = \prod_{v \in V} P(X_v \mid X_{pa(v)})$ [18], [21].

3.3 Related Bayesian Net Models

Bayesian nets have been used in various studies in software engineering area. Most of them have been focused on predicting development effort and/or software quality.

The BN model developed by Beaver [5] is the most relevant to this study because it explicitly refers to ISO 9126 standard. However, the author does not provide the full model structure or any details on defined probability distributions. Thus, it is difficult to reuse, adapt or validate this model.

Wagner [38] proposed a framework for building BNs for software quality prediction. It assumes using the activity-based quality models. The author performed a validation for maintainability prediction. However, it is not clear if it is possible or sensible to follow this approach to build models for integrated quality prediction, i.e. where there are links between quality features.

Fenton et al. [8] developed a BN model for trade-off analysis between software functionality, project effort and achieved software quality. In this model, quality is reflected by two variables – a more objective defect rate and a typically subjective customer satisfaction. Radliński et al. [28] implemented a similar concept of trade-offs in a structurally new model where the quality is expressed as number of defects and defect rate. The authors of both models performed their internal validation using hypothetical scenarios.

The analysis of existing BN models have also involved studies focused on a single quality aspect, such as maintainability [36] or defects [1], [7], [9]. This study also takes advantages of other BNs, such as those focused on development effort prediction, which we have discussed in detail in our recent paper [30]. Although there are very few direct relationships between most of these models and detailed software quality modeling, we have still exploited the knowledge and experience gained when building and validating previous models.

4 A Framework for Integrated Software Quality Prediction

Proposed framework is 'aware' that the resulting model is a Bayesian net. Thus, it covers the activities specifically focused on building a BN model, subject to specific requirements related with areas of software development and software quality modeling. Fig. 1 illustrates an overview of the framework using notation of the UML activity diagram. This framework consists of five main phases:

1. Preparation – this involves problem definition and gathering expert knowledge and empirical data. Gathering expert knowledge may be informal (e.g. loose interviews) or more structured (e.g. questionnaires). Gathering empirical data includes data analysis and interpretation of results.
2. Defining model structure – this covers five tasks performed sequentially starting with defining a list of main-level features and terminating with defining additional links. Detailed model structure depends on the outcome of tasks in phase 1.
3. Questionnaire survey – this covers preparing the questionnaire and performing the survey, including analysis of results. The contents of the questionnaire depend on the model structure.

4. Defining distributions for variables – this involves encoding the results from questionnaire survey, other expert knowledge and empirical data as the probability distributions for variables.
5. Validating the model – this involves validating the whole model. The type of validation depends on specific situation and user needs. Typically, the model is analyzed for correctness, i.e. if expert knowledge and/or empirical data are encoded correctly, and for accuracy, i.e. if the model provides accurate predictions.

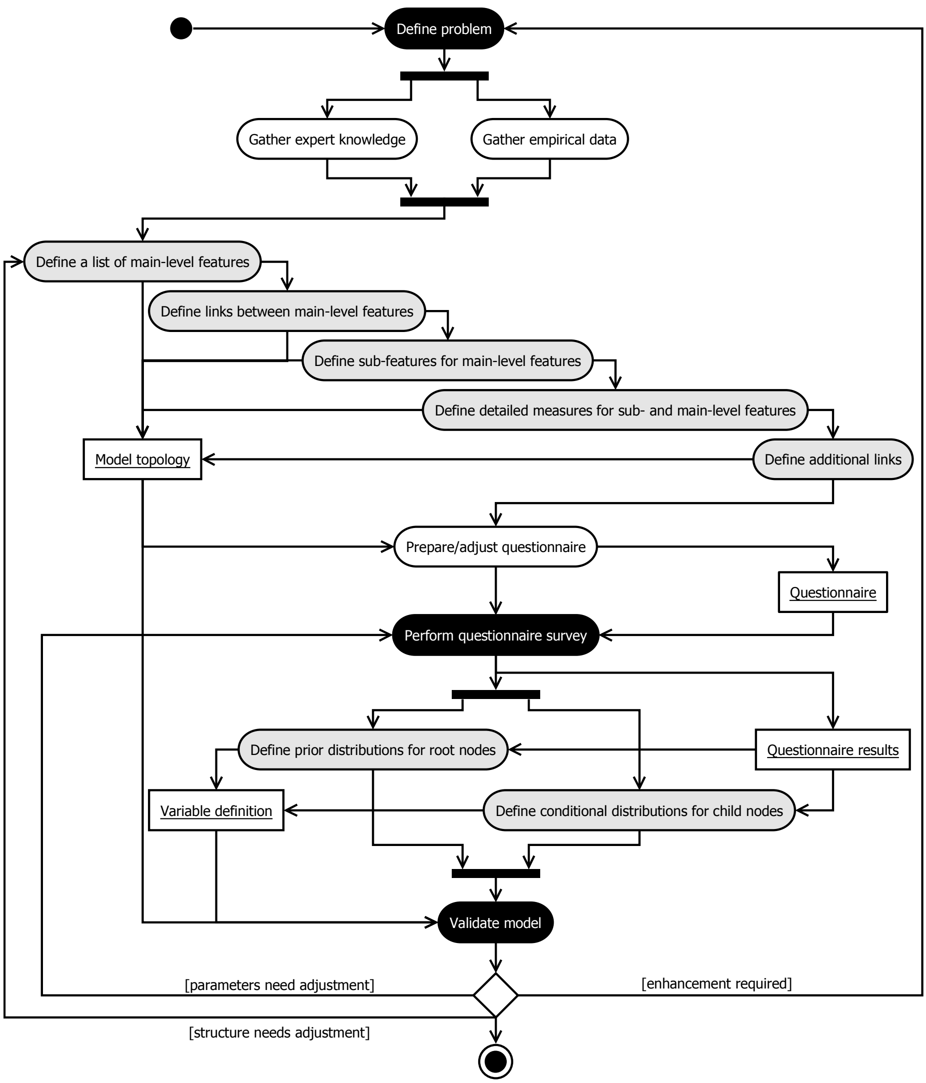


Fig. 1. An overview of proposed framework – milestone tasks highlighted with dark background, tasks directly related with development of predictive model with grey background

It is possible to perform the validation after each phase or even each task, especially if the tasks and model are large and complex. In addition, it is possible to perform some groups of tasks iteratively until reaching specific target. To avoid illegibility Fig. 1 does not show such detailed paths and loops.

Depending on particular needs, the framework may be used at various levels of details for software quality. These include module, package, component, application or system containing a set of applications. However, since the process of building a model is time consuming, the framework may be inefficient for smaller software parts such as modules or packages. On the other hand, through the reuse of the model, gathered knowledge and data, the framework may still be cost-effective.

5 Bayesian Net Model Developed with Proposed Framework

5.1 Knowledge Base for Building the Bayesian Net

Following the framework proposed in previous section, we have developed a Bayesian net for integrated software quality prediction. The framework assumes performing the questionnaire survey to gather the expert knowledge and then encoding it in the model. Because we have not completed the survey yet, we have built this model based on developed knowledge base. This knowledge base reflects expert knowledge and empirical data published in software quality literature as well as scientific and industrial experience of author and collaborators. We have used the following main sources to develop specific knowledge/model area:

- selection of variables – features and sub-features: [11], [12], [13], [14], [15], [16];
- links between features: [17], [19], [20], [25], [35];
- selection of variables – quality measures: [1], [4], [17], [19], [20], [25], [35];
- probabilistic definition of variables: expert judgment, based on [17], [19], [20], [25], [26], [32], [35], [37];
- controllable factors: expert judgment, based on existing models [7], [9], [28].

Fig. 2 summarizes high-level relationships between software quality features encoded into the model. The '+' sign indicates the positive relationship where the increased level of one feature will likely *cause* the increased level of another feature. The '–' sign indicates the negative relationship, where the increased level of one feature will likely *cause* the decrease in the level of another feature, unless additional compensation, such as effort, will be provided. The blank cell indicates no significant relationship between a pair of features. We have defined these links based on the literature mentioned above and other expert knowledge. They should be treated as an example – in a specific environment it is possible to define them differently, according to the needs of specific project and its environment.

	functional suitability	reliability	security	compatibility	operability	performance efficiency	maintainability	portability	usability	safety
reliability	+									
security										
compatibility		+	–							
operability	+	+		+						
performance efficiency			–	–	–					
maintainability	+	+			+	–				
portability				+	+	–	+			
usability	+	+	+		+	–	+			
safety	+	+	+	+		–			+	
flexibility	+	+	–		+	–	+	+	+	–

Fig. 2. An example of relationships between quality features

To make the model usable in decision support we have incorporated some controllable variables. These controllable factors reflect the amount of effort allocated to specific phases and the process quality in specific phases. We assume that they have a positive impact on the following quality features:

– requirements effort and requirements process quality → functional suitability, operability, maintainability, usability, flexibility;
– implementation effort and implementation process quality → functional suitability, reliability, performance efficiency, maintainability, usability, flexibility;
– testing effort and testing process quality → reliability, compatibility, performance efficiency, usability, safety.

The assumption for this study is that effort and process variables have strong impact on relevant quality features, i.e. stronger than the links among quality features. We should point that the impact of effort on software quality is not deterministic. It means that the increase of effort causes the *expected*, but not guaranteed, increase of software quality.

5.2 Model Structure

Developed Bayesian Net for Integrated Software Quality Prediction (BaNISoQ) consists of four sets (layers) of variables: features, sub-features, measures and controllable variables. Fig. 3 illustrates the part of the structure for the BaNISoQ model. Due to the model complexity, it is not possible to show all variables in this paper. Full structure is available online at [31].

In the highest layer, there are quality features, each with its own indicators – sub-features. In the second layer there are sub-features; each of them contains its own set of measures. These measures form the lowest level. Thus, each quality feature is a hierarchical naïve Bayesian model. Such structure enables flexible adding, removing and calibrating variables at lower levels, depending on availability of expert knowledge or empirical data. There are links between relevant quality features to reflect positive or negative relationships between features. Additionally, selected features are children of some controllable variables. These links reflect the impact of process factors on particular quality features.

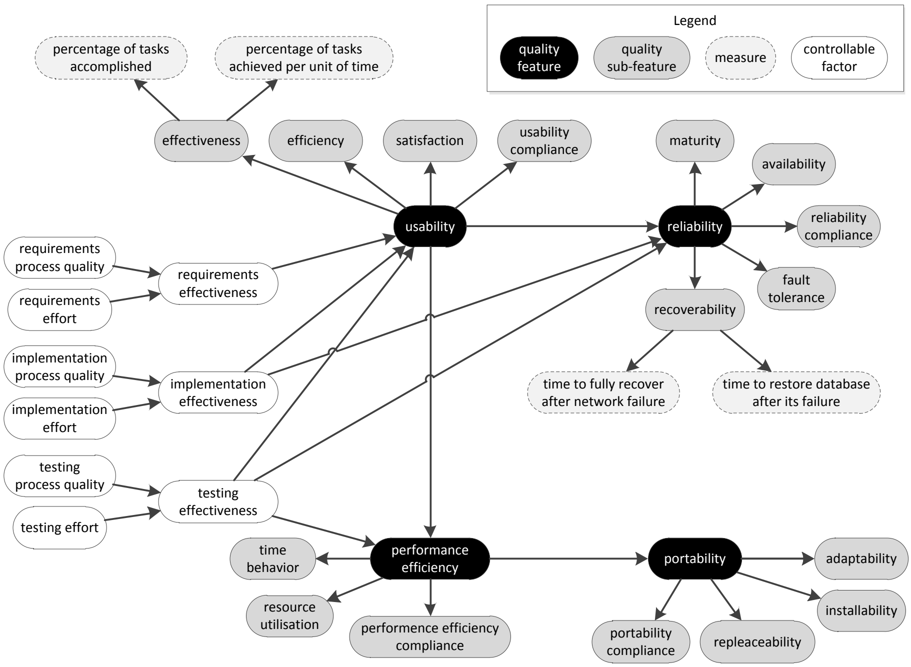


Fig. 3. Part of the BaNISoQ model

All features, sub-features and process variables are defined on a ranked scale with five states from 'very low' to 'very high' and probability distributions in the form of expressions. These definitions exploit the functionality available in AgenaRisk BN tool [2], where the ranked variable is internally a numeric variable over the range (0, 1). This range is divided into a set of five equal-length intervals such as (0, 0.2) for 'very low', (0.2, 0.4) for 'low' and so on. This enables using various types of expressions that are usually easier to elicit, more informative, and easier to adjust in comparison with manually filling the probability tables. It also enables calculating summary statistics for predicted variables as for typical numeric variables.

Table 1 provides examples of expressions for selected variables. We have defined them using truncated normal distribution (TNormal) with the following parameters:

1. Mean – typically referring to the parent variable (e.g. in sub-features) or more complex expression performed on a set of parent variables (e.g. in features).
2. Variance – for features has been set to 0.05, for sub-features as different values from 0.01 (strong relationship with parent variable) to 0.1 (medium relationship), for some controllable variables to 0.001 (very strong relationship).
3. Left truncation point – the value at which the left side of distribution is truncated, for ranked variables always set to 0.
4. Right truncation point – the value at which the right side of distribution is truncated, for ranked variables always set to 1.

The ‘wmean’ denotes weighted mean function with even number of parameters. The odd-numbered parameters are the values of weights for the even-numbered parameters.

Table 1. Definition of selected variables

Type	Variable	Definition
feature	usability	TNormal(wmean(1, 0.5, 3, wmean(3, *req_effect*, 2, *impl_effect*, 1, *test_effect*), 1, *funct_suit*), 0.1, 0, 1)
feature	reliability	TNormal(wmean(1, 0.5, 4, wmean(2, *impl_effect*, 3, *test_effect*), 2, wmean(1, *funct_suit*, 1, *maintain*, 1, *usability*, 1, *flexibility*)), 0.05, 0, 1)
sub-feature	effectiveness	TNormal(*usability*, 0.01, 0, 1)
sub-feature	usability compliance	TNormal(*usability*, 0.05, 0, 1)
sub-feature	efficiency	TNormal(*usability*, 0.1, 0, 1)
measure	percentage of tasks accomplished	*effectiveness* = ’very high’ → Normal(95, 10) *effectiveness* = ’high’ → Normal(90, 40) *effectiveness* = ’medium’ → Normal(75, 60) *effectiveness* = ’low’ → Normal(65, 80) *effectiveness* = ’very low’ → Normal(50, 100)
controllable	testing effort	TNormal(0.5, 0.05, 0, 1)
controllable	testing process quality	TNormal(0.5, 0.05, 0, 1)
controllable	testing process effectiveness	TNormal(wmean(3, *test_effort*, 4, *test_procq*), 0.001, 0, 1)

The BaNISoQ model does not provide the aggregated value of software quality, but it is possible to add it to the model. This overall quality may be defined using weighted expressions, such as WMEAN, WMAX or WMIN [10]. It would aggregate the levels of quality features with appropriate weights, depending on particular analysis environment. We will investigate it in future.

6 Validation

The main aim for the validation stage was to analyze if the model correctly encodes relationships between relevant variables (simulations 1 and 2). This involved investigating predicted probability distributions and summary statistics. Due to space constraints, we present results for calculated mean values for predicted variables – using the fact that the underlying scale for ranked variables is (0, 1) as explained in Section 5.2. The third simulation reveals the model potential for decision support.

The first simulation has been focused on validating correctness of links among different quality features. Here, we analyzed the impact of each quality feature in two runs. In the first run, we assigned the state ‘medium’ to a particular feature, in the second run – a state ‘very high’.

For the first run, with the state ‘medium’ assigned to any quality feature, the predicted mean value for all other features was 0.5 (analyzed to 15 decimal places). This shows that the model provides correct baseline estimates.

Table 2 illustrates the results for the second run. Values above 0.5 indicate the increase of feature level caused by setting an observation ‘very high’ to a single feature. For example, the first value in row ‘reliability’ indicates that setting reliability to ‘very high’ instead of default distribution causes the shift of mean value for functional suitability from 0.5 to 0.55. Values above 0.5 in Table 2 correspond to marks ‘+’ in Fig. 2 while values below 0.5 in Table 2 correspond to marks ‘–’ in Fig. 2. It means that the model correctly encodes the relationships from knowledge base.

Table 2. Predicted mean values for quality features in simulation 1

Predicted Observed	func.	relia.	secu.	comp.	oper.	perf.	main.	port.	usab.	safe.	flex.
functional suit.		0.54			0.52		0.54		0.56	0.52	0.54
reliability	0.55			0.53	0.51		0.55		0.55	0.53	0.54
security				0.47		0.47			0.52	0.51	0.48
compatibility		0.53	0.47		0.52	0.48		0.54		0.53	
operability	0.52	0.51		0.52		0.47	0.52	0.54	0.53		0.53
performance eff.			0.47	0.48	0.47		0.47	0.46	0.48	0.49	0.48
maintainability	0.54	0.54			0.52	0.47		0.54	0.54		0.54
portability				0.53	0.54	0.46	0.54				0.54
usability	0.56	0.54	0.52		0.53	0.48	0.54			0.52	0.54
safety	0.52	0.52	0.51	0.53		0.49			0.52		0.49
flexibility	0.54	0.54	0.48		0.53	0.48	0.54	0.54	0.54	0.49	

The second simulation has been focused on analyzing the correctness of links between controllable factors (allocation of effort and process quality) and software quality features. This simulation, as the first one, also involved two runs. In the first run, we have assigned a state ‘medium’ to each controllable factor (one by one). Predicted mean value for all quality features was 0.5 (analyzed to 15 decimal places), so the model provides correct baseline estimates.

In the second run we have assigned a state ‘very high’ to each controllable factor (also one by one) and we have investigated the levels of predicted quality features. Table 3 illustrates the results of this simulation run – the interpretation of values is the same as for simulation 1. The predicted mean values for all investigated quality features are above 0.5. It means that the increase of development effort causes the increase of the level of all relevant quality features, as assumed in knowledge base.

The values in Table 3 deviate more from the baseline value 0.5 than in Table 2. It means that the relationships between effort and quality features are stronger than among quality features. Thus, the model correctly encodes this assumption.

The third simulation has been focused on illustrating how the model may be used for decision support in a hypothetical situation. Let us suppose that some knowledge about an ongoing project is available. Due to the project complexity, ‘high’ amount of effort in requirements and implementation phases is required. After entering this information, the model predicts the levels of quality features. Fig. 4 (top) illustrates two of them.

Let us assume that for a project manager predicted level of reliability is unacceptably low and the desired level is ‘high’. The model provides revised predictions and says that it is possible to achieve such target by increasing the amount of testing effort and testing process quality (Fig. 4 middle).

Table 3. Predicted mean values for quality features in simulation 2

Scenario Predicted feature	requirements effort='VH'	implementation effort='VH'	testing effort='VH'
functional suitability	0.54	0.54	
reliability		0.55	0.55
security			
compatibility			0.54
operability	0.57		
performance efficiency		0.52	0.52
maintainability	0.54	0.54	
portability			
usability	0.54	0.53	0.51
safety			0.53
flexibility	0.54	0.53	

However, the manager is aware that the project is under time pressure and they may allocate only 'low' level of testing effort. When entering this observation to the model and relaxing the one for reliability, the model provides further revised prediction for reliability and usability (Fig. 4 bottom), which are expected to be even lower than in the baseline scenario. This simulation demonstrates high potential for various types of analysis for which the model may be used for, including goal-seeking, trade-off analysis, causal and diagnostic inference.

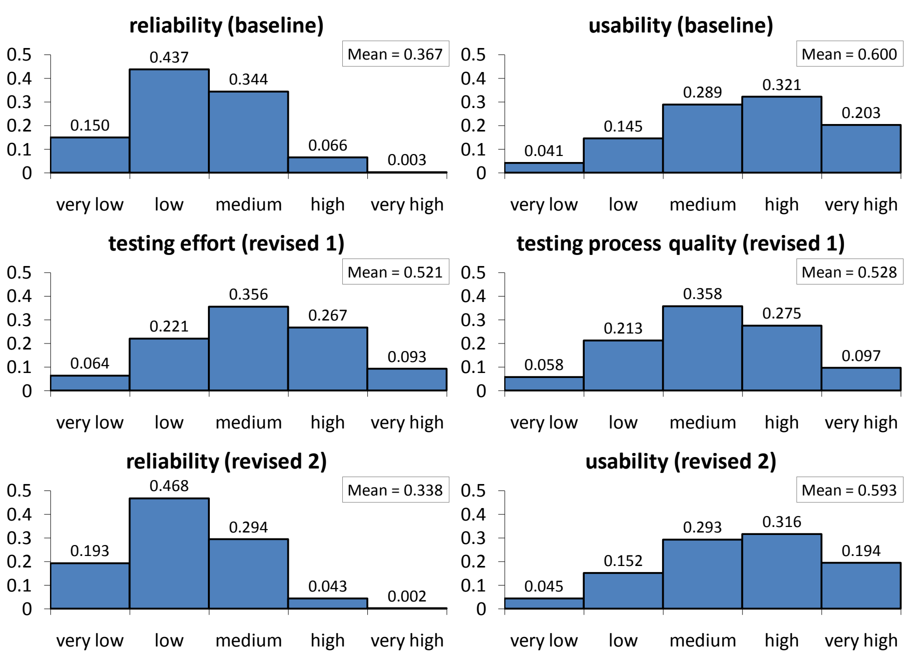


Fig. 4. Results of predictions for simulation 3

7 Applications in Other Fields

The focus of both the proposed framework and the developed predictive model is on the area of software quality. With some adjustments this framework may be used in modeling other types of problems and domains. This may be possible because the constraints defined for the framework do not directly refer to the specific issues of software quality modeling. Rather, they structure the process of developing the predictive model and the model itself.

This framework may be of an interest in situations where:

- the problem under investigation is complex but can be divided to a set of subproblems,
- there is no or not enough empirical data to generate a reliable model from them,
- domain expert (or group of experts) are able to define, calibrate and enhance the model,
- the relationships are of stochastic and non-linear nature,
- there is a need for a high analytical potential involving causal and evidential reasoning, and optimization – in combination with external techniques.

Application of the framework in other fields may not be straightforward though. Specifically, it may be difficult when, apart from the stochastic relationships, there are also strong deterministic relationships that need to be reflected with high precision (for example in areas such as physics and engineering).

8 Limitations and Threats to Validity

The proposed approach has a high analytical potential and it may be used in research and industrial applications. However, it still has some limitations and threats related to its validity.

First, Bayesian nets enable reflecting causal relationships between variables of interest. Currently this framework only partially supports the definition of causal relationships, mainly between quality features. Thus, the model may not reflect all relationships that exist in reality because it is essentially a set of linked naïve Bayes models (which do not contain explicit causal relationships). It is possible to add some causal links between leaf variables to have a set of tree augmented naïve Bayes models.

Second, the model enables static analysis, i.e. for a specific point of time. In reality, the domain of software engineering is more complex because of dynamic changes. These changes are reflected in development process and software evolution. Proposed framework and model support a basic form of 'dynamics' – as a set of specifically defined variables. Converting this approach to so-called Dynamic Bayesian Nets is significantly more complex and we will investigate it in future.

Third, the model proposed in this paper does not attempt to be a general model for predicting software quality in every possible environment. Because there are various points of view on software quality and various models of software quality, it is not possible to satisfy all of them in a single predictive model. Our approach rather focuses on interactive model definition by the end user by providing basic frames for this purpose. It is the point of view of particular user, which will be reflected in the final model. Specific quality features may be outside the area of interest in some cases, and thus may not be included in the model.

For similar reasons some quality features have a larger number of observable variables associated while other features have very few observable variables. This is additionally related with the fact that some quality features are generally more difficult to define, estimate and predict.

Further, although the framework simplifies the definition of distributions by using expressions, in some cases the model may still be difficult to calibrate. For example, this happens when defining a nominal variable, which has a parent of numeric variable. This definition requires answering questions such as: what is the probability that the nominal variable *X* will have a state ‘x’ if the numeric variable *Y* has the value ‘y’? It may be difficult to answer such unintuitive questions even for experts in statistics or artificial intelligence. We plan to simplify the process of calibration by providing supporting software tool using Bayesian inference and graphical definition of the relationships.

Finally, typically there is an expectation to validate developed models for predictive accuracy and compare it with other models. We have not performed such validation in this study. As explained earlier, this is related with the lack of appropriate validation dataset and the lack of similar type of predictive models. Such integrated prediction of software quality requires high volume of empirical data describing various details of software project and the process of its development. We plan to perform a questionnaire survey among software experts and use the results of this survey as validation dataset.

9 Conclusion and Future Work

Presented framework for building BN model for integrated software quality prediction may simplify the process of building such BNs. This may be especially important for domain experts with no or little experience with BN modeling.

We have used this framework to develop the Bayesian Net model for Integrated Software Quality prediction (BaNISoQ). The ‘integration’ of this model is mainly related with incorporation a variety of software quality features and relationships between them. Performed validation of this model revealed its high analytical potential.

Both the proposed framework and the predictive model have some limitations. In future we will focus on overcoming them. We believe that ultimately, proposed framework and the BN model may be a central part of future intelligent system for enhanced software quality analysis and decision support. To improve the usability and accuracy of the approach, we will also focus on extending it by automatic extraction of software and process data from software repositories. Additionally we plan to extend the approach to reflect the architecture and hierarchy of analyzed software by taking into account the diversity and differences between parts of software.

Acknowledgments. I would like to thank Norman Fenton from Agena, Ltd. in UK for providing the Agenarisk BN tool. This work has been supported by research funds from the Ministry of Science and Higher Education in Poland as a research grant for years 2010-2012.

References

1. Abouelela, M., Benedicenti, L.: Bayesian Network Based XP Process Modelling. Int. J. Softw. Eng. Appl. 1, 1–15 (2010)
2. AgenaRisk BN Tool, Agena (2009), `http://www.agenarisk.com`
3. Akiyama, F.: An Example of Software System Debugging, In: Federation for Information Processing Congress, Ljubljana, vol. 71, pp. 353–379 (1971)
4. Alvaro, A., Santana De Almeida, E., Romero De Lemos Meira, S.: A software component quality framework. SIGSOFT Softw. Eng. Not. 35, 1–18 (2010)
5. Beaver, J.M.: A life cycle software quality model using bayesian belief networks, Doctoral Dissertation, University of Central Florida Orlando, FL (2006)
6. Côté, M.-A., Suryn, W., Georgiadou, E.: In search for a widely applicable and accepted software quality model for software quality engineering. Softw. Qual. J. 15, 401–416 (2007)
7. Fenton, N., Hearty, P., Neil, M., Radliński, Ł.: Software Project and Quality Modelling Using Bayesian Networks. In: Meziane, F., Vadera, S. (eds.) Artificial Intelligence Applications for Improved Software Engineering Development: New Prospects, pp. 1–25. Information Science Reference, New York (2008)
8. Fenton, N., Marsh, W., Neil, M., Cates, P., Forey, S., Tailor, M.: Making Resource Decisions for Software Projects. In: 26th Int. Conference on Software Engineering, Washington DC, pp. 397–406 (2004)
9. Fenton, N., Neil, M., Marsh, W., Hearty, P., Radliński, Ł., Krause, P.: On the effectiveness of early life cycle defect prediction with Bayesian Nets. Empir. Softw. Eng. 13, 499–537 (2008)
10. Fenton, N.E., Neil, M., Caballero, J.G.: Using Ranked Nodes to Model Qualitative Judgments in Bayesian Networks. IEEE Trans Knowl. Data Eng. 19, 1420–1432 (2007)
11. ISO/IEC 25000:2005, Software Engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE (2005)
12. ISO/IEC FDIS 25010:2011, Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models (2011)
13. ISO/IEC FDIS 9126-1:2001, Software Engineering – Product quality – Part 1: Quality model (2001)
14. ISO/IEC TR 9126-2:2003, Software engineering – Product quality – Part 2: External metrics (2003)
15. ISO/IEC TR 9126-3:2003, Software engineering – Product quality – Part 3: Internal metrics (2003)
16. ISO/IEC TR 9126-4:2004, Software Engineering – Product quality – Part 4: Quality in use metrics (2004)
17. Jarvis, A., Crandall, V.: Inroads to Software Quality: "How to" Guide and Toolkit. Prentice Hall PTR, Upper Saddle River (1997)
18. Jensen, F.V., Nielsen, T.D.: Bayesian Networks and Decision Graphs, 2nd edn. Springer, Heidelberg (2007)
19. Jones, C.: Applied Software Measurement: Global Analysis of Productivity and Quality, 3rd edn. McGraw-Hill, New York (2008)
20. Kan, S.H.: Metrics and Models in Software Quality Engineering. Addison-Wesley, Boston (2003)
21. Kjærulff, U.B., Madsen, A.L.: Bayesian Networks and Influence Diagrams. A Guide to Construction and Analysis. Springer, New York (2008)

22. Lyu, M.: Handbook of software reliability engineering. McGraw-Hill, Hightstown (1996)
23. Mair, C., Kadoda, G., Lefley, M., Phalp, K., Schofield, C., Shepperd, M., Webster, S.: An investigation of machine learning based prediction systems. J. Syst. Softw. 53, 23–29 (2000)
24. Musa, J.D.: Software Reliability Engineering: More Reliable Software Faster and Cheaper, 2nd edn. Author house, Boston (2004)
25. O'Regan, G.: A Practical Approach to Software Quality. Springer, New York (2002)
26. Ortega, M., Perez, M., Rojas, T.: Construction of a Systemic Quality Model for Evaluating a Software Product. Softw. Qual. J. 11, 219–242 (2003)
27. Pfautz, J., Koelle, D., Carlson, E., Roth, E.: Complexities and Challenges in the Use of Bayesian Belief Networks: Informing the Design of Causal Influence Models. In: Human Factors and Ergonomics Society 53rd Annual Meeting, San Antonio, pp. 237–241 (2009)
28. Radliński, Ł., Fenton, N., Neil, M., Marquez, D.: Improved Decision-Making for Software Managers Using Bayesian Networks. In: 11th IASTED Int. Conf. Software Engineering and Applications, pp. 13–19. IASTED, Cambridge (2007)
29. Radlinski, L., Hoffmann, W.: On Predicting Software Development Effort using Machine Learning Techniques and Local Data. Int. J. Softw. Eng. Comput. 2, 123–136 (2010)
30. Radlinski, L.: A Survey of Bayesian Net Models for Software Development Effort Prediction. Int. J. Softw. Eng. Comput. 2, 95–109 (2010)
31. Radliński, Ł.: BaNISoQ: Bayesian Net Model for Integrated Software Quality Prediction (2010), http://lukrad.univ.szczecin.pl/projects/banisoq/
32. Rosqvist, T., Koskela, M., Harju, H.: Software Quality Evaluation Based on Expert Judgement. Softw. Qual. J. 11, 39–55 (2003)
33. Rubey, R.J., Hartwick, R.D.: Quantitative measurement of program quality. In: ACM National Conference, pp. 671–677. ACM, New York (1968)
34. Schneidewind, N.F.: An approach to software reliability prediction and quality control. In: Proc. of the American Federation of Information Processing Societies, pp. 337–347. ACM, New York (1972)
35. Schulmeyer, G.G., McManus, J.I. (eds.): Handbook of Software Quality Assurance. Prentice Hall PTR, Upper Saddle River (1999)
36. Van Koten, C., Gray, A.R.: An application of Bayesian network for predicting object-oriented software maintainability. Inf. Softw. Technol. 48, 59–67 (2006)
37. Villalba, M.T., Fernández-Sanz, L., Martínez, J.J.: Empirical support for the generation of domain-oriented quality models. IET Softw. 4, 1–14 (2010)
38. Wagner, S.: A Bayesian network approach to assess and predict software quality using activity-based quality models. In: 5th Int. Conf. on Predictor Models in Software Engineering. ACM Press, New York (2009)
39. Zhang, D., Tsai, J.J.P.: Machine Learning and Software Engineering. Softw. Qual. J. 11, 87–119 (2003)

Process Instance Management Facilities Based on the Meta Process Models

Seunghun Park[1], Sangyoon Min[1], and Doohwan Bae[2]

[1] Division of SPI Consulting, SolutionLink Co., South Korea
[2] Department of Computer Science, College of Information Science & Technology, Korea Advanced Institute of Science and Technology

Abstract. In the field of software process management, many studies have proposed a variety of process management technologies. However, most of the process management technologies have treated managerial analysis facilities for software process instances too lightly. It results in less attention from industry. To overcome the problem, we propose the process instance management facilities in the structural and behavioral aspects based on the meta process models. The meta process models consist of the two types of models: meta structural model and the meta behavioral model. Based on the meta process models, a process model is developed and two types of process instance models are generated using the process model: the structural instance model and the behavioral instance model. For the structural instance model, we adopt process slicing. On the other hands, we use several analysis techniques for the behavioral instance model. The proposed approach enables a project manager to analyze structural and behavioral properties of a process instance and allows a project manager make use of the formalism for the management facilities without knowledge for the formalism.

1 Introduction

In principle, that the quality of software is governed by the quality of the software process, has been widely received as a fundamental premise in software engineering. Under this principle, many studies have been done to improve a software process in various areas such as software process management, software process maturity framework, software process simulation, and so on [1,3,5,8,9,11,21,23,24]. In the field of software process management, the software engineering community has been exerting great effort since Osterweil argued "Software processes are software, too" [16].

It is no doubt that the existing works have made great contributions in advancing process management technology in many aspects. Many studies, however, haven't received much attention from industry, although they are technically successful. One of the causes is the lack of managerial analysis facilities for a software process instance, which is a project-level software process. Project managers have to make many decisions about a software process before and after the start of a project. For example, when a software process is changed, a

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 326–341, 2011.

project manager needs to answer several questions: "Which parts of the software process should be changed", "Will the process change improve the performance of a project", and so on. Project managers may want to deal with behavioral management such as schedule delay in a process instance as well as structural management for constituents of a software process (e.g. activities and agents).

We propose the structural and behavioral management facilities for a software process instance based on the meta process models. The meta process models work as predefined templates for instantiating process entities and consist of the meta structural model and the meta behavioral model to capture different aspects of a software process in separate ways. Based on the meta process models, a process model is developed using graphical user interfaces. When developing the process model is completed, two types of process instance models are generated: the structural instance model and the behavioral instance model. We suggest adopting process slicing to structurally manage a process instance. Process slicing can provide an opportunity to analyze change impacts on a software process instance by identifying the scope of changes [19,20]. On the other hands, we provide analysis techniques related to the efficiency of conducting activities for the behavioral management of a process instance. These techniques are derived from a timed Petri-net formalism based on the meta behavioral model. The usefulness of our approach is validated by applying it to real projects. The result shows that our approach is applicable to real projects and helpful to make a decision.

The proposed approach has several advantages. First, the behavioral management facility enables a project manager to analyze the activity duration and human resource utilization during process execution, which is one of the key concerns of a project manager, using the timed Petri-net formalism. Second, the meta process model allows a project manager take advantage of the formalism for the behavioral management facility in a non-intrusive way. User can model a software process through making instances from the metal process model constructs without the need to knowing the complex modeling formalism. Third, the structural and behavioral management facilities give an opportunity to a project manager to more accurately estimate the time and effort required to make changes in a process instance.

The remainder of this paper is organized as follows. Section 2 introduces related work, and Section 3 describes the method to develop process instance models based on the meta process model and the managerial techniques focusing on the behavioral management facilities. Section 4 shows a case study to validate our approach. Section 5 summarizes the main results of this paper and gives a plan for future work.

2 Related Work

Many researchers have focused on process modeling, analysis, and execution [2,4,10,12] since the introduction of the pivotal idea, process programming by Leon Osterweil [16]. However, most of them neglect providing the managerial analysis facilities for project managers. They provide only primitive analysis techniques supported by the modeling formalism they are using (e.g. deadlock and trap detection in Petri-net based approach).

Several studies provide the techniques to identify the scope of change during a process change [6,7,25,26]. They identify the elements of a software process or workflow affected by proposed changes and produce a list of entities that should be addressed during the proposed change process. Dependency analysis is a representative technique to analyze the scope of change based on the relationships between the constituents. It examines the constituents on the paths from or to the source of change by traversing the paths. These approaches can help a project manager make a decision for structural changes on a software process, but they can only cover the structural management facility on the software process.

The software process simulation technique is a useful tool to analyze and demonstrate the effects of changes in the process instance level and decision-making by quantitatively representing the behavioral observations of a process instance. Many studies are proposed techniques or models to support the analysis for a variety of management issues in software projects such as strategic management, planning, control, prediction, and process improvement [8,9,21]. Those represent such a process as currently implemented (as-is) or as planned for future implementation (to-be). The simulation technique can aid to decision making and reduce risks in software projects. However, in many cases the simulation technique requires the detailed skills for simulation tools or languages as well as knowledge for designing the process to be performed; it results in a burden for project managers or developers [17,18,22].

3 Management Facilities for a Process Instance

Figure 1 shows the overview of our approach. The fundamental concept of our approach is based on the hierarchical structure of a software process [15]. We introduce the concept of the meta process models, which are generic but formal process model specifications predefined at meta level. A user can model a software process through making instances from the metal process models constructs. To relieve the user from burdening process modeling effort, we provide a PERT-chart like graphical user interface. As soon as the modeling is done, the modeled process is transformed into the process instance models. The process instance models represent actual processes instantiated from the meta process models and consist of the structural instance model and the behavioral instance model corresponding to the meta process models. The structural instance model is transformed into a process dependency model to apply process slicing. The behavioral instance model contains the formalism for the behavioral property of a software process. Several analysis techniques are applied based on the formalism. We do not describe how to transform the structural instance model into the process dependency model because it is the same as the mechanism in previous literatures [19,20]. Instead, we focus on the behavioral management techniques developed upon the meta behavioral model.

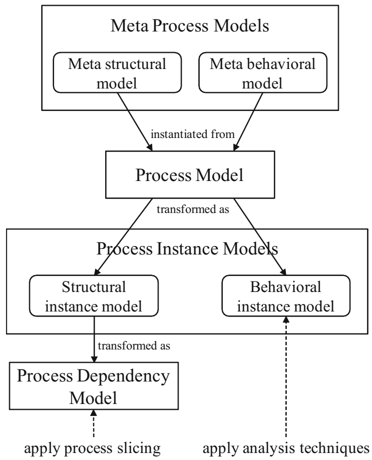


Fig. 1. Overview of our approach

3.1 Meta Process Models

When we develop the meta process models, we consider a software process as software. Thereby, we can specify a software process in different aspects (i.e. structural and behavioral) as we model ordinary software. The meta process models consist of the meta structural model and the meta behavioral model.

The structural aspect of a software process is defined by its constituent entities and their relationships to each other. The types of major constituent entities of a software process are activity, artifact and agent. The meta structural model captures these basic process entities and their relationships in meta level. Figure 2 shows the meta structural model. A software process consists of a set of activities, artifacts, and agents. An activity may have three kinds of reflexive relationships: sequential, selective, and synchronous parallel [15,20]. The sequential relationship means that the target activity can start only after the source

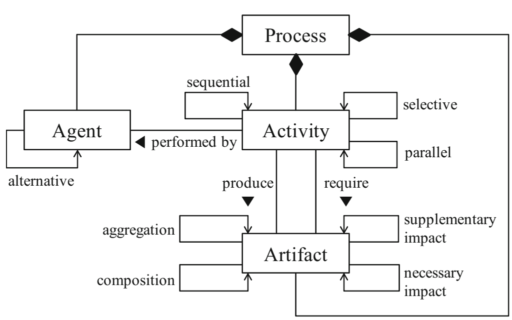


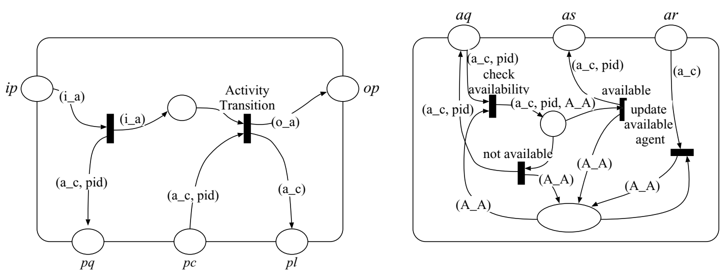
Fig. 2. Meta structural model

activity has finished. This relationship does not allow parallelism between the two activities. The selective relationship means that the target activity can be started depending on the result from finishing the source activity. The synchronous parallel relationship means that the target activity can be finished after the source activity has finished. This relationship allows the two activities to be performed simultaneously. An artifact may have four kinds of reflexive relationships: aggregation, composition, necessary impact relationship, and supplementary impact relationships [20]. The aggregation relationship means that an artifact is a part of another artifact. This relationship specifies that the lifetime of the part artifact is independent of the lifetime of the whole artifact. The composition relationship also means that an artifact is a part of another artifact. However, this relationship specifies that the lifetime of the part artifact is dependent on the lifetime of the whole artifact. The necessary impact relationship indicates that the source artifact is a necessary input for producing the target artifact. The relationship implies that the target artifact is certainly affected by the change in the source artifact. On the other hand, the supplementary impact relationship indicates that the source artifact is a supplementary input for producing the target artifact. An agent may have an alternative relationship between agents [20]. This relationship means that one agent may be able to perform activities assigned to the other agent, and vice versa. The relationship between activity and agent captures the assignment relationships between agents and activities. There are two kinds of relationships between the activity and the artifact. An artifact may be used as input or output of the activity.

The meta behavioral model defines execution behavior of a software process which is determined by its pre and post conditions of activities, and order of its execution. The preconditions (e.g. the existence of input artifacts and agent) and post conditions (e.g. existence of output artifacts and release of agent) of an activity can be specified for various elements. The meta behavioral model is defined by Modeling, Analysis, and Management net (MAM-net) [13] which is a Petri-net-based process modeling formalism based on Predicate Transition net (Pr/T net). MAM-net provides high-level modeling constructs that specify the behaviors of the process entities. The complete formal definition of MAM-net can be found in the previous literature [13]. We provide a concise definition of MAM-net as follows:

Definition 1: *MAM-net M*
$M = \{PS, AAU, IP, OP, PI, AI, T, F, P, A, M_0\}$, where

- $PS = \{ps_1, ps_2, ..., ps_n\}$ is a finite set of processSteps. A processStep represents a process activity supporting hierarchical abstraction. Figure 3(a) depicts the internal structure of an atomic processStep.
- $AAU = AgentAllocationUnit$ is an agent allocation unit which represents a pool of the currently available human agents. Figure 3(b) depicts the internal structure of an AgentAllocationUnit.
- $IP = \{ip_1, ip_2, ..., ip_n\}$ is a finite set of input-artifact-receiving ports of processSteps



(a) Internal Structure of Atomic ProcessStep (b) Internal Structure of AgentAllocationUnit

Fig. 3. Internal Structure of Atomic processStep and AgentAllocationUnit

- $OP = \{op_1, op_2, ..., op_n\}$ is a finite set of output-artifact-sending ports of processSteps. Each processStep has a single input-artifact-receiving port and a single output-artifact-sending port for transferring artifacts.
- $PI = \{pq, pc, pl\}$ is a set of agent allocation ports of processSteps. Each processStep has three communication ports; one is for sending requests for human agents, another for receiving the requested agents, and the other for releasing the finished agents. These ports are called *pq*, *pc*, and *pl*, respectively. In MAM-net, each processStep must communicate with the AgentAllocationUnit for agent allocation.
- $AI = \{aq, as, ar\}$ is a set of process agent allocation ports of the AgentAllocationUnit. AgentAllocationUnit also has three communication ports corresponding to PI. We named these three ports aq, as, and ar. The port aq receives agent requests from the port *pq*. The port as is the port for sending the requested agents to the port *pc*. The port ar receives the released agents from the port *pl* of processSteps.
- $T = \{t_1, t_2, ...t_m\}$ is a finite set of transitions.
- F is a finite set of arcs.
- P is a set of places.
- A is a set of annotations for triggering functions.
- M_0 is initial marking denoting the availability of initial artifacts and agents.

3.2 Process Model

Based on the meta process models, a process model is developed to specify all necessary information for the deterministic transformation into the corresponding instance models. We adopt *cognitive-level modeling*, which provides graphical user interfaces such that user can model a software process using a PERT chart-like interface [14]. As shown in Fig. 4, the two modeling constructs are a process activity and an artifact flow to reduce user difficulty as much as possible.

While the representation is simple, the activity contains several hidden properties specified in the meta process model related to the actual process as follows: a sequential set, a selective set, a parallel set, inputs, outputs, assigned agents, alternative agents. Similarly, the artifact contains several hidden properties as

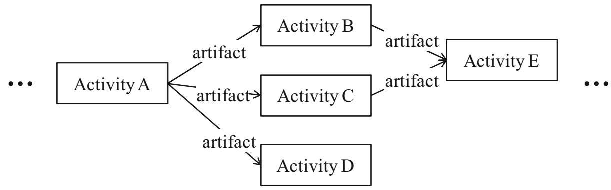


Fig. 4. Process model template

follows: a composition set, an aggregation set, a direct impact set, an indirect impact set.

3.3 Process Instance Models

Once the process model is developed, the user-defined information contained the process model is transformed into the formal specifications, which are process instance models. Process instance models consists of two types of instance models: the structural instance model and the behavioral instance model.

The structural instance model is defined as follows:

Definition 2: *Structural instance model SIM*
$SIM = \{A_c, A_r, A_g, R_{actf}, R_{artf}, R_{alt}, R_{AG}, R_{AR}\}$, where

- $A_c = \{a_1, a_2, a_3, ..., a_i\}$ is a finite set of activity nodes representing the actual activities instantiated from the entity *Activity* in the meta structural model.
- $A_r = \{r_1, r_2, r_3, ..., r_j\}$ is a finite set of artifact nodes representing the actual artifacts instantiated from the entity *Artifact* in the meta structural model.
- $A_g = \{g_1, g_2, g_3, ..., g_k\}$ is a finite set of agent nodes representing the actual agents instantiated from the entity *Agent* in the meta structural model.
- R_{actf} is a set of relationships between activity instances specified in the meta structural model.
- R_{artf} is a set of relationships between artifact instances specified in the meta structural model.
- R_{alt} is a set of relationships between agent instances.
- R_{AG} is a set of relationships between activity and agent instances
- R_{AR} is a set of relationships between activity and artifact instances.

Based on the definition of the structural instance model, it can be obtained by mapping the properties of the process model to the constituents of the structural instance model. Figure 5 shows an example of the structural instance model.

The behavioral instance model is developed by transforming the process model information into MAM-net. A process model is transformed into a top-level MAM-net. Each activity is transformed into a processStep and a relationship between activities is transformed into a transition between processSteps according to the transformation rules as shown in Fig. 6.

After all the activities and relationships are transformed, AgentAllocationUnit is assigned to the top-level MAM-net. Figure 7 shows an example of the behavioral instance model.

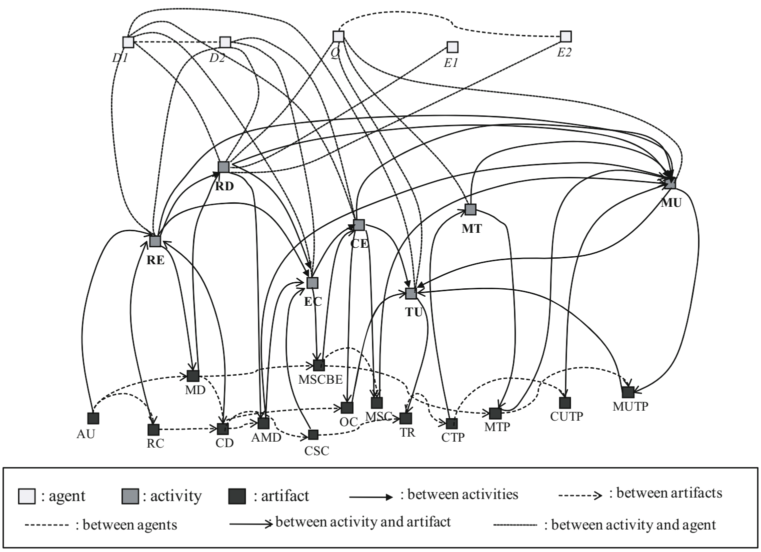


Fig. 5. Example of a structural instance model

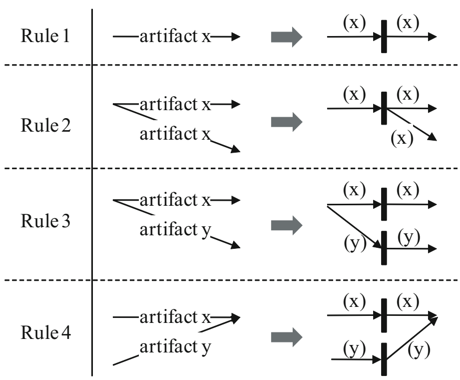


Fig. 6. Transformation rules for artifact flows

3.4 Behavioral Management Techniques for a Process Instance

There are many managerial elements related to the efficiency of conducting activities. In our approach, we focus on the elements related to activity duration and human resource utilization during process execution. MAM-net offers effective facilities for examining the following properties which are frequently considered by project managers in managerial decision making.

- Cumulative time consumption of each process activity
- Spots of agent conflict under a given schedule
- Influence of agent conflict

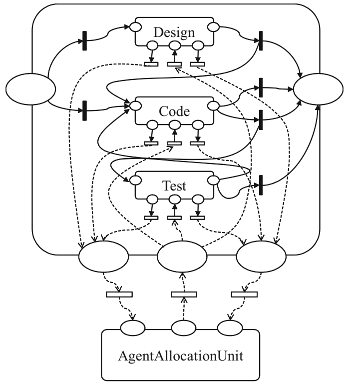


Fig. 7. Example of the behavioral model

- Agent utilization
- Artifact idle time

In the analysis of these properties, we assume that the expected duration of each process activity is determined during the project planning.

The managerial analysis mechanism is derived from timed Petri-net formalism. Based on the firing behavior of timed Petri net, we analyze the time-related behavior of MAM-net. In MAM-net, the delayed transitions are the activity transitions of atomic processSteps. Thus, in MAM nets, the activity transition of each processStep has a positive delay time, and all the other transitions have zero delay time. For our managerial analysis, we give some definitions, and derive equations for the time-related behavior of processSteps.

Let TC denote the time consumption of its argument. Let the activity transition of an atomic processStep ps_i be denoted by t_{ps_i}. Each atomic processStep has exactly one single activity transition. Therefore, the time consumption of an atomic processStep is equal to the time consumption of its activity transition and can be defined as follows:

$$TC(ps_i) = TC(t_{ps_i}) \tag{1}$$

In the managerial analysis, we analyze the time-related behaviors of the process model, considering the agent allocation(i.e. agent conflict). We then compute the properties using the analysis result. Before further derivation, we define the following attributes of the atomic processStep:

- $Input(ps_i)$: the set of input artifacts of processStep ps_i
- $Output(ps_i)$: the set of output artifacts of processStep ps_i
- $Prec(ps_i)$: the set of preceding processSteps of processStep ps_i
- $Input.arrival(ps_i)$: the latest arrival time among the input artifacts at ip_{ps_i}
- $Agent.request(ps_i)$: the time when ps_i makes the agent request through its pq_{ps_i}
- $Agent.arrival(ps_i)$: the time when ps_i receives the requested agent through its pc_{ps_i}

– $Agent.release(ps_i)$: the time when ps_i releases the requested agent through its pl_{ps_i}
– $Start(ps_i)$: the execution start time of processStep ps_i
– $Finish(ps_i)$: the execution finish time of processStep ps_i
– $Agent(ps_i)$: the set of agents that ps_i needs for its execution
– $Require(a_i)$: the set of processSteps that need an agent a_i for its execution
– $Conflict.candidate(ps_i)$: the set of processSteps that have potential agent conflicts with ps_i, such that $(Agent(ps_i) \wedge Agent(ps_k)) \neq \emptyset$ for all $ps_k \in Conflict.candidate(ps_i)$

The latest arrival time of input artifacts for processStep ps_i is the latest output producing time of its preceding processSteps. Thus, we define the latest arrival time of the input artifacts for processStep ps_i, $Input.arrival(ps_i)$, as follows:

$$Input.arrival(ps_i) = Max\{Finish(ps_{pc}) \mid ps_{pc} \in Prec(ps_i)\} \tag{2}$$

The activity transition in each atomic processStep is constrained by not only its input artifacts but also by the arrival of the requested agents.

Since each processStep ps_i sends an agent-request token as soon as all necessary input artifacts arrive, $Agent.request(ps_i)$ is defined as follows:

$$Agent.request(ps_i) = Input.arrival(ps_i) \tag{3}$$

The execution of processStep ps_i starts only after both the necessary input artifacts and requested agent arrive. Thus, the start time of the actual execution of ps_i is defined as follows:

$$Start(ps_i) = Max\{Input.arrival(ps_i), Agent.arrival(ps_i)\} \tag{4}$$

Therefore, the execution finish time of processStep ps_i is redefined as (5):

$$Finish(ps_i) = Max\{Input.arrival(ps_i), Agent.arrival(ps_i)\} + TC(ps_i) \tag{5}$$

Considering agent conflicts, we can define the $Agent.arrival(ps_i)$ as follows:

$$Agent.arrival(ps_i) = Max[Agent.request(ps_i), Max\{Finish(ps_k) \mid ps_k \in Conflict.set(ps_i)\}] \tag{6}$$

When we consider the agent conflict, there can be two types of agent conflict. One is where the requested agents were already allocated to other processSteps. The other is where multiple processSteps are requesting the same agent simultaneously. In the second case, a managerial scheduling policy is required. We use the shortest-job-first policy. Possible conflicts can be detected using (4) and $Conflict.candidate(ps_i)$. We define the set of conflict processSteps with a processStep ps_i as follows:

$$Conflict.set(ps_i) = \{ps_k \in Conflict.cadidate(ps_i) | Start(ps_k) \leq Start(ps_i)\} \tag{7}$$

Furthermore, the attributes *Start* and *Agent* of each processStep in $Conflict.set(ps_i)$ describe when and to whom the conflict occurs.

Using (4), (5), and (6), the conflict-free start time and finish time of each processStep are defined as follows:

$$\begin{aligned} Start(ps_i) = Max[&Input.arrival(ps_i), \\ &Max[Agent.request(ps_i), \\ &Max\{Finish(ps_k) \mid ps_k \in Conflict.set(ps_i)\}]] \end{aligned} \tag{8}$$

$$\begin{aligned} Finish(ps_i) = Max[&Input.arrival(ps_i), \\ &Max[Agent.request(ps_i), \\ &Max\{Finish(ps_k) | ps_k \in Conflict.set(ps_i)\}]] + TC(ps_i) \end{aligned} \tag{9}$$

Using (8) and (9), we can produce a conflict-free process schedule, and find the influence of possible agent conflicts.

Agent utilization in the context of our approach implies the work hours of each agent divided by the total duration of the entire process activities. Thus, it is defined as (10):

$$Utilization(a_i) = \frac{\sum_{all\ ps_k \in Require(a_i)} TC(ps_k)}{Max\{Finish(ps_i) \mid ps_i \in PS\}} \tag{10}$$

The idle time of an artifact means the waiting time of the artifact before consumed by other activities due to agent conflicts. Because the output artifact of each processStep is transfered to succeeding processSteps no matter what agent conflicts occur, the idle time of each artifact is the difference between the arrival time of the artifact and the actual start time of the consuming processStep. Thus, the idle time of each artifact can be measured using (11).

$$\begin{aligned} Idle(art_k) = &Start(ps_i) - Finish(ps_j) \\ &where\ ps_j \in Prec(ps_i) \wedge ((art_k \in Output(ps_j)) \wedge \\ &(art_k \in Input(ps_i))) \end{aligned} \tag{11}$$

4 Case Study

4.1 Background

As the experimental study, we use the project planning and tracking data of the actual industry projects in an IT organization. The IT organization is the IT department of a commercial bank consisting of about 150 software engineers in South Korea. We have randomly selected 10 software projects that have been conducted in one year. Most projects vary in their project duration from one month to a few months. We explain the results and reasoning the causes of the results as well as lessons learned.

4.2 Experimental Results

Schedule. For each activity of the selected projects, we analyzed the predicted delays, and compared them to the actual delays of the corresponding activities. Figure 8 shows the relations between them each dot on the graph represents an activity. X axis is the analyzed delay in days. Y axis is the actual delay in days.

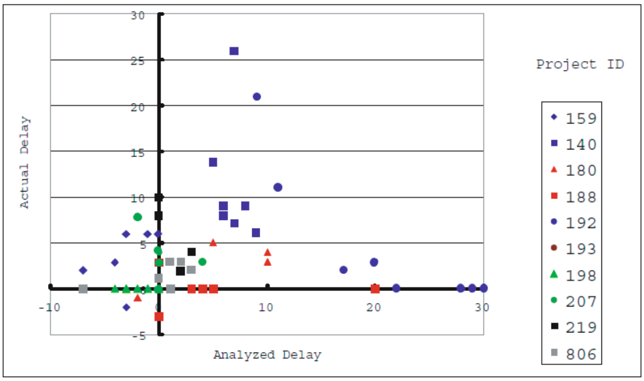


Fig. 8. Analyzed Delays of Activities v.s. Actual Delays of Activities

As you can notice the graph shows a positive regression relation between the analyzed delay and the actual delay, although some special cases exist. The dots on the X axis are the last activities that have tendency to finish the project no matter how much delay they have in the preceding activities. Majority of the dots on the Y axis is the ones with bad estimation for coding activity. The delay of the coding activity causes delays of the succeeding activities in chain reaction. The project 198 whose activities lie on the negative X axis is a special case that all activities are finished on time. Actually we found that this project intentionally had loose schedule. The project participants delayed the activity just until their exact due dates. Another special case is the project 188 who activities lie on the positive X axis. All activities of the project finished on time for some reason which we could not figure out analytically. The one possible reason for such phenomenon is the overtime working by the project members.

We found that the analyzed numerical result of each activity delay is not meaningful if we compare the analyzed delay of each activity with the actual delay of the corresponding activity. However, the analyzed tendency for delay seems to have meaningful results. Total 79 activities exits in the 10 projects as shown on Fig. 8. If we discard the starting activity and final activity for each project, that are usually not affected by the analysis logic in our approach, we have 59 internal activities left. If we discard the extreme case projects(i.e. project 188 and 198), we have 42 activities left. Among them, 28 activities have the similar tendency in their actual execution comparing to the analyzed delay. For the 28 activities(which is 66 percent of 42 activities), if analyzed result shows delay, the actual execution was delayed.

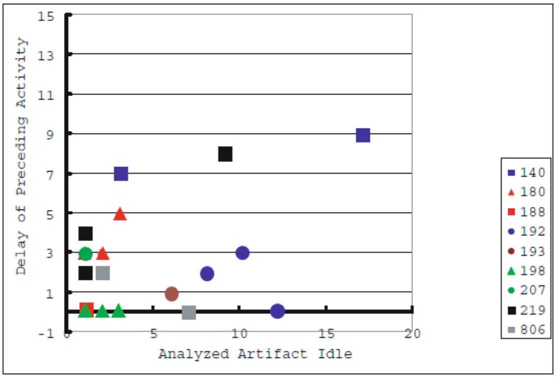


Fig. 9. Analyzed Artifacts Idle Times v.s. Delays of the Preceding Activities

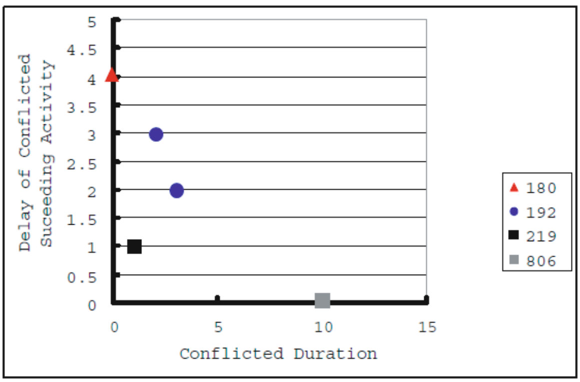


Fig. 10. Analyzed Durations of Agent Conflicts v.s. Delays of the Preceding Activities

Artifacts Idle Time. One of the observations we found through the empirical study is that the practitioners have tendency to delay the work that is not in the critical path. The activities which produce the artifacts with the idle time are logically not in the critical path. In this section, we provide empirical observation on the analyzed artifact idle time v.s. the delay of the preceding activities.

As shown in Fig. 9, we can graphically observe that positive relations between the idle times of the artifacts and the delays of the activities that produce the artifacts. In our experimental study, we found 24 artifacts that have the idle times greater than zero. Among the corresponding preceding activities, 16 activities have delay times greater than zero in their actual execution which is around 67 percent out of 24 activities.

Agent Conflicts. In a logical sense, if there exists an agent conflict occurred between two activities that have overlapped duration and use the same agent, at least one of the activity must be delayed. In the empirical study, we found 7 agent conflicts in there planned process. A single agent conflicts has two conflicted

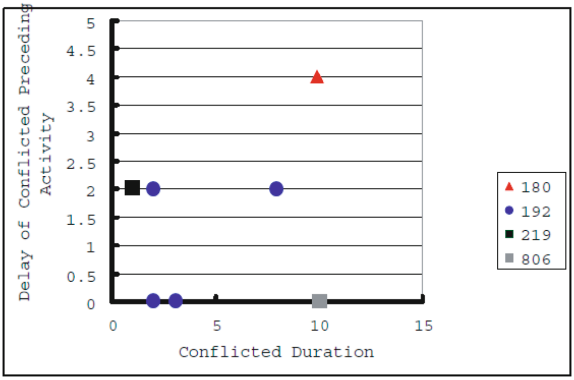


Fig. 11. Analyzed Durations of Agent Conflicts v.s. Delays of the Succeeding Activities

activities which are overlapped to each other. Thus, the total number of the activities under the agent conflicts is 14. In the experimental study we found 8 activities among those activities have the delay times greater than zero, in their actual execution. 8 out of 14 is around 57 percent. This number is close to the half of the total activity, which shows positive evidence to our reasoning. Figures 10 and 11 show the relation between the analyzed duration of agent conflicts and the delay of the preceding and the succeeding activities that are overlapped and under the conflicts.

5 Conclusion

Although many researches have been proposed in the software process technology, most of them haven't be adopted in industry. The most important issue we believe is the management facilities for a in project-level software process. To resolve the issue, we proposed an approach to providing structural and behavioral management facilities for a software process instance based on the meta process model. We suggest the three-level modeling framework: the meta level, the model level, and the instance level. Based on the concept of the meta level (i.e. the meta process model), we can adopt the structural management technique, process slicing, and the behavioral management techniques such as agent conflicts. The result of the empirical study shows that our approach is applicable to real projects and helpful to make a decision. Project managers can improve the planned process by analyzing the activity dependency before the execution of the process. Finding out potential delays of the planned activities can give project managers the chance to revise the project process with less deviation against actual project performance. In similar concept, analyzing possible agent conflicts can improve the planned process. In real world, we often face with the situation that a certain project participant is heavily allocated to a project. In many situation, such participants usually perform multiple roles at the same time during the project life cycle.

As future research, we have a plan to develop more managerial techniques. Developing some predefined standard processes library that are compliant to the maturity frameworks such as CMMI and ISO 15504 may be helpful for the technology adoption by industry. Automated gathering and measurement of process quality metrics would be beneficial. Integration with cost estimation techniques such as the COCOMO II would be very useful.

Acknowledgments. This work was partially supported by Defense Acquisition Program Administration and Agency for Defense Development under the contract.

References

1. Ambriola, V., Conradi, R., Fuggetta, A.: Assessing process-centered software engineering environments. ACM Trans. Software Engng. Methodol. 6, 283–328 (1997)
2. Avrilionis, D., Cunin, P., Fernstrom, C.: OPSIS: a view mechanism for software processes which supports their evolution and reuse. In: Proceedings of the 18th International Conference on Software Engineering, Berlin, Germany, pp. 38–47 (1996)
3. Bandinelli, S., Nitto, E., Fuggetta, A.: Policies and mechanisms to support process evolution in PSEES. In: the 3rd International Conference on the Software Process, Reston, VA, USA, pp. 9–20 (1994)
4. Bandinelli, S., Fuggetta, A., Ghezzi, C., Lavazza, L.: SPADE: an environment for software process analysis, design, and enactment. In: Software Process Modeling and Technology, pp. 223–247. Research Studies Press (1994)
5. Curtis, B., Kellner, M.I., Over, J.: Process modeling. Communications of the ACM 35(9), 75–90 (1992)
6. Chou, S., Chen, J.J.: Process program change control in a process environment. Software Practice and Experience 30(3), 175–197 (2000)
7. Dai, W., Covvey, D., Alencar, P., Cowan, D.: Lightweight query-based analysis of workflow process dependencies. Journal of Systems and Software 82(6), 915–931 (2009)
8. Dapeng, L., Qing, W., Junchao, X.: The role of software process simulation modeling in software risk management: A systematic review. In: the 3rd ESEM, Lake Buena Vista, Florida, pp. 302–311 (2009)
9. Donzelli, P.: A Decision Support System for Software Project Management. IEEE Software 23(4), 67–75 (2006)
10. Doppke, J., Heimbigner, D., Wolf, A.: Software process modeling and execution within virtual environment. ACM Trans. Software Engng Methodol. 7, 11–40 (1998)
11. Guo, Y., Seaman, C.: A survey of software project managers on software process change. In: Proceedings of ESEM 2008, Kaiserslautern, Germany, pp. 263–269 (2008)
12. Jaccheri, M.L., Conradi, R.: Techniques for process model evolution in EPOS. IEEE Transactions on Software Engineering 19(12), 1145–1156 (1993)
13. Min, S., Bae, D.: MAM nets: A Petri-net based approach to software process modeling, analysis and management. In: The 9th International Conference on Software Engineering and Knowledge Engineering, Madrid, Spain, pp. 78–86 (1997)

14. Min, S.: A software process modeling framework based on meta process model. Ph.D. Dissertation, Korea Advanced Institute of Science and Technology, KAIST (2002)
15. OMG, Software Process EngineeringMetamodelSpecification. Version 2.0, OMG-Document formal/2008-04-01 (2008)
16. Osterweil, L.: Software processes are software too. In: The 9th International Conference on Software Engineering,Washington, DC, pp. 2–13 (1987)
17. Park, S., Choi, K., Yoon, K., Bae, D.: Deriving software process simulation model from SPEM-based software process model. In: The APSEC 2007, Nagoya, Japan, pp. 382–389 (2007)
18. Park, S., Kim, H., Kang, D., Bae, D.: Developing a software process simulation model using SPEM and analytical models. International Journal of Simulation and Process Modeling 4(3-4), 223–236 (2008)
19. Park, S., Kim, H., Bae, D.: Change impact analysis of a software process using process slicing. In: QSIC 2009, Jeju, Korea, pp. 359–366 (2009)
20. Park, S., Bae, D.: An approach to the software process change impact using process slicing and simulation. Journal of Systems and Software 84, 528–543 (2011)
21. Pfahl, D., Ruhe, G.: IMMoS: A Methodology for Integrated Measurement, Modelling and Simulation. Software Process Improvement and Practice 7(3-4), 189–210 (2002)
22. Raffo, D., Nayak, U., Wakeland, W.: Implementing Generalized Process Simulation Models. In: ProSim 2005, St. Louis, Missouri, pp. 139–143 (2005)
23. Rus, I., Neu, H., Munch, J.: A Systematic Methodology for Developing Discrete Event Simulation Models of Software Development Process. In: ProSim 2003, Portland, Oregon (2003)
24. SEI: Capability Maturity Model Integration V1.3 (2010)
25. Soffer, P.: Scope analysis: identifying the impact of changes in business process models. Software Process Improvement and Practice 10(4), 393–402 (2005)
26. Sun, P., Jianga, C.: Analysis of workflow dynamic changes based on Petri net. Information and Software Technology 51(2), 284–292 (2009)

A Multi-paradigm Complexity Metric (MCM)

Sanjay Misra, Ibrahim Akman, and Ferid Cafer

Faculty of Engineering, Department of Computer Engineering,
Atilim University, Ankara, Turkey
{smisra,akman,fcafer}@atilim.edu.tr

Abstract. Huge amount of researches and software metrics have been proposed for procedural and object-oriented languages. However, there are only few metrics available in the literature related with multi-paradigm programming languages. In this paper, we propose a metric to evaluate the code written in multi-paradigm language. Our proposed metric can be used for most of the programming paradigms, including both procedural and object-oriented languages.

Keywords: Software complexity, complexity metrics, Python, software development.

1 Introduction

Software quality has been raising demand for decades due to its complexity. Software development processes are treated as a complex task, therefore to control its complexity and to maintain the quality of software is a challenging job. A software product should carry several quality attributes, such as correctness, reliability, efficiency, integrity, usability, maintainability, testability, flexibility, portability, reusability, and interoperability [1]. According to Somerville [2] the most necessary software quality attribute is maintainability and, to be able to maintain a software system efficiently, the codes should be understandable for the developers. This means, to achieve high quality code, reduction of code complexity is essential. Metrics are indicators of complexity and can be used to estimate software quality. There are a number of metrics each focusing on different complexity factors [3].

One may find the hundreds of metrics in the literature for evaluating the software quality. These metrics [4-7, 18-25] are of several types and for different purposes. Somerville [2] categorizes metrics as control and predictor metrics. Control metrics are related to software processes, whereas predictor metrics are associated with software products. Control metrics estimate effort, time and defects and predictor metrics assess the number of attributes and structures in a code. The literature provides several metrics for procedural and OO paradigms. However, there are very few metrics for multi paradigm languages [26-27]. This constitutes our motivation and the present paper proposes a metric for multi paradigm languages, which combines the features of procedural and OO languages.

Generally, code complexity depends on its external and internal characteristics which can be characterized by functionality and efficiency respectively. In order to

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 342–354, 2011.

include the complexity due to functionality and due to code efficiency the proposed metric includes Function Point and the cognitive complexity.

The paper is organized in the following way. The classifications of languages are given in section 2. The proposal of the metric is given in section 3. The metric is demonstrated in section 4. The conclusion drawn on this work is in section 5.

2 Classification of Languages

Programming languages are based on programming paradigms. Under the scope of the research the programming languages are classified as;

Procedural programming: Procedural programming languages [15], which are also known as imperative programming languages, are based on structures and structural flow of algorithms. Programming language C is a good example to procedural languages.

In the past, researchers proposed their methodologies for evaluating codes, which were written in procedural languages [10, 11], such as C.

OO programming: In the early 80's studies focused on OO programming languages, [8, 9]. It provides data abstractions of hierarchical classes for programmers. Java is a popular [12] example for programming languages in this category.

Some of the benefits of OO are faster development, higher quality, easier maintenance, reduced costs, increased scalability, improved structures, and higher level of adaptability [13].

Multi-paradigm programming: A multi-paradigm language may support two or more programming paradigms and provides their advantages. A problem may require various concepts to be solved practically and therefore usage of multi-programming paradigm [16] may be needed. One of the main benefits of a multi-paradigm language is providing easier transitions between paradigms [15].

In fact, in recent years, the multi-paradigm approach is becoming popular for software development amongst the programmers [14]. Probably, C++ is the best example to multi-paradigm languages and is known as the most popular in software development applications in the last decade [12].

3 Proposed Metric

Although, all stages in the development life-cycle need to be evaluated from the quality point of view it is usually expected that the most important one is the quality of the code, which is highly affected by the programming paradigms. Therefore, metrics used for measuring code complexity can be classified as procedural metrics, OO metrics, and multi-paradigm metrics according to the above classification of paradigms. The proposed metric is based on the characteristics of multi-paradigm codes.

Under the scope of the present proposal, by multi-paradigm, the combination of procedural and OO paradigms is meant. In this case, it encompasses characteristics of both procedural and OO paradigms and, therefore, a programmer may choose to use either procedural or OO paradigm, or even a combination of them in the same code.

As discussed in the introduction, it is generally expected that a quality metric should consider all important factors in its programming paradigm in order to provide a complete understanding of the code quality and code complexity. In this respect, the proposed metric is developed to consider all factors responsible for internal and external complexities of procedural and OO programming features. Internal complexity corresponds to multi-paradigm complexity measurement, whereas external complexity corresponds to Function Point (FP). Accordingly, the proposed measure for the code quality of a multi-paradigm program is given by;

1. **Code Quality:** The quality of code in multi-paradigm language is defined as:

$$CQ = (FP / MCM) \qquad (1)$$

where, CQ is Code Quality, FP is the Function Point calculation for the code, and MCM is the multi-paradigm complexity measurement which computes the complexity value for all factors responsible for OO and/or procedural part of the code.

It is easy to see from the formula that the proposed metric CQ is a function of Function Points (FP) and the complexity value of all other factors responsible for the code. This is because, external and internal quality factors are not completely independent. Generally, the lower MCM value has a positive impact on FP and increases total code quality. This means that developing cognitively less complex code yields more functional and more efficient program. Therefore, proposed metric obtains code quality by dividing FP by MCM. With this characteristic, the proposed metric can provide better and more reliable understanding for the tasks whose internal code qualities are the same but external qualities are different or opposite.

In the computation of CQ, the division of FP by MCM usually gives an extremely small number and this may be difficult to make decisions on. Therefore, experiences showed that it is appropriate to multiply the division by 10,000 in order to make the result easy to understand and compare. Thus the code quality is defined by

$$\mathbf{CQ = (FP / MCM) * 10{,}000} \qquad (1.1)$$

2. **Function Point [17]:** To compute function points, there are 14 questions to be answered:

    1. Does the system require backup and recovery?
    2. Are the specialized data communications required to transfer information to or from the application?
    3. Are there distributed processing functions?
    4. Is performance critical?
    5. Will the system run in an existing, heavily utilised operational environment?
    6. Does the system require on-line data entry?
    7. Does the on-line data entry require the input transaction to be built over multiple screens or operations?
    8. Are the ILFs updated on-line?
    9. Are the inputs, outputs, files, or inquiries complex?
    10. Is the internal processing complex?
    11. Is the code designed to be reusable?

12. Are conversion and installation included in the design?
13. Is the system designed for multiple installations in different organisations?
14. Is the application designed to facilitate change and for ease of use by the user?

Above questions should be numbered from 1 (lowest) to 5 (highest).

After answering above questions, the following Table 1 should be filled:

Table 1. Function Point

Information Domain Value	Weighting factor				
	Count	**Simple**	**Average**	**Complex**	**Total**
EIs	x	3	4	6	
EOs	x	4	5	7	
EQs	x	3	4	6	
ILFs	x	7	10	15	
EIFs	x	5	7	10	
Count Total					

EIs: External Inputs

EOs: External Outputs

EQs: External Inquiries

ILFs: Internal Logical Files

EIFs: External Interface Files

$$FP = \text{count total} \times [0.65 + 0.01 \times E(F_i)] \quad (2)$$

3. Multi-paradigm Complexity Measurement (MCM)

As mentioned before, MCM is the integration of all the factors, which are responsible for increasing the complexity of code. MCM is a modified version of the approach proposed by [18], which presents a metric for Python language to provide an extension for multi-paradigm complexity measurement.

As noted before MCM is the sum of complexity of OO and procedural part of the code and defined as follows:

Multi paradigm Complexity Measurement (MCM) = Complexity (factors effecting the OO features + factors effecting the procedural part of the code)

A complete code of a system normally consists of classes and the main program. Further, all the features of OO codes are confined on the classes. So it will be better to estimate the complexity of all the classes, which contains different type of OO features. Normally the classes are either in inheritance hierarchy or in distinct classes. Accordingly, the multi-paradigm complexity measurement is formulated as;

$$MCM = CIclass + CDclass + Cprocedural \tag{3}$$

where,

CIclass = Complexity of Inherited Classes
CDclass = Complexity of Distinct Classes
Cprocedural = Procedural Complexity

These factors are defined as follows:

To calculate the CIclass, CDclass and Cprocedure, first we have to calculate the complexity of all classes as independent classes, which further will be placed in inheritance hierarchy or as distinct classes.

The complexity of an independent class is given by

$$Cclass = W(attributes) + W(variables) + W(structures) + W(objects) - W(cohesion) \tag{3.1}$$

where, Cclass = Complexity of a single class.

The reason of subtraction of cohesion is that it reduces the complexity and thus it is desirable [17].

where, weight of attributes or variables is defined as [18]:

$$W(variables\ or\ attributes) = 4 * AND + MND \tag{3.1.1}$$

where, AND = Number of Arbitrarily Named Distinct Variables/Attributes and
MND = Number of Meaningfully Named Distinct Variables/Attributes

Weight of structure W(structures) is defined as:

$$W(structures) = W(BCS) \tag{3.1.2}$$

where, BCS are basic control structure.

Weight of objects, Weight(objects) is defined as:

$$W(objects) = 2 \tag{3.1.3}$$

Creating an object is counted as 2, because while creating an object constructor is automatically called. Thus, coupling occurs. Therefore, it is the same as calling a function or creating an object. Here it is meant to be the objects created inside a class. Moreover, a method that calls another method is another cause of coupling, but that fact is added to MCM value inside Weight (structures).

Weight of cohesion is defined as [17]:

$$W(cohesion) = MA / AM \tag{3.1.4}$$

Where, MA = Number of methods where attributes are used and
AM = Number of attributes used inside methods

While counting the number of attributes there is no any importance of AND or MND.

Based on the above information CIclass can be defined as;

There are two cases for calculating the complexity of the Inheritance classes depending on the architecture:

- If the classes are in the same level then their weights are added.
- If they are children of a class then their weights are multiplied due to inheritance property.

If for a certain root class there are m levels of depth in the OO code and level j has n classes then the Cognitive Code Complexity (CCC) [25] of the system is given as

$$CIclass = \prod_{j=1}^{m} \left[\sum_{k=1}^{n} CC_{jk} \right] \tag{3.2}$$

CDclass can be defined as;

$$CDclass = Cclass(x) + Cclass(y) + \ldots \tag{3.3}$$

The reader should note that all classes, which are neither inherited nor derived from another, are parts of CDclass even if they have caused coupling together with other classes.

Cprocedural can be defined as;

$$Cprocedural = W(variables) + W(structures) + W(objects) - W(cohesion) \tag{3.4}$$

weight of variable W(variable) is defined as:

$$W(variables) = 4 * AND + MND \tag{3.4.1}$$

The variables are defined globally.

Weight of structure W(structures) is defined as:

$$W(structures) = W(BCS) + object.method \tag{3.4.2}$$

where, BCS are basic control structure, and those structures are used globally. ‘object.method’ is used when procedural code is calling a reachable method of a class using an object. If the program consists of only procedural code, then the weight of the ‘object.method’ will be 0.

Weight of objects W(objects) is defined as:

$$W(objects) = 2 \tag{3.4.2}$$

Creating an object is counted as 2, as it is described above (2.3). Here it is meant to be the objects created globally or inside any function which is not a part of any class. If the program consists of only procedural code, then the weight of the ‘objects’ will be 0.

$$W(cohesion) = NF / NV \tag{3.4.3}$$

where, NF is number of functions, and NV means number of variables. Coupling is added inside W (structures) as mentioned in the beginning of the description of the metric.

3 Demonstration of the Metric

We demonstrate the code complexity metric through a case study. This case study includes 8 classes. The complete program of this case study is given in Appendix. The program consists of classes of shapes written in C++. The root class is Shapes and other classes, Figure1P, Square, Circle, Figure2P, Rectangle, Oval and Colour are derived from it. Colour class is an outside of the inheritance, but its method is called by other classes.

According to the proposed metric Cclass, CIclass, CDclass, Cprocedural values of the system are calculated. It is worth to mention that during the calculation of complexity of inheritance, CIclass should be carefully computed. The complexity of the classes at the same level should be added and those values should be multiplied with their parent classes, as shown in the following computation.

First we compute the MCM for this program. The components of MCM for all classes are summarized in Table 2. The procedural complexity of the system is summarized in Table 3.

Table 2. Class Complexity of Shapes in C++

class	att	str	var	obj	MA	AM	Cohesion	Comp.
Colour	0	33	2	0	0	0	0	35
Shapes	2	6	0	0	2	2	1	7
Figure1P	1	8	0	0	2	1	2	7
Square	0	27	0	2	0	0	0	29
Circle	0	27	0	2	0	0	0	29
Figure2P	1	11	0	0	1	1	1	11
Rectangle	0	27	0	2	0	0	0	29
Oval	0	27	0	2	0	0	0	29

Table 3. Procedural Complexity of Shapes in C++

Non-Class	var+str+obj	Complexity
Cprocedural	32	32

CIclass = Shapes * (Figure1P * (Square + Circle + Figure2P * (Rectangle + Oval)))

= 7 * (7 * (29 + 29 + 11 * (29 + 29)))

= 34104

CDclass = 35

Cprocedural = 32
MCM = CIclass + CDclass + Cprocedural
= 34104 + 35 + 32
= 34171

We have calculated MCM now we estimate the function points. The Information domain values and their weights are given in Table 4.

Table 4. Function Point Calculation of Shapes

Information Domain Value	Weighting factor				
	Count	**Simple**	**Average**	**Complex**	**Total**
EIs	1	3	**4**	6	4
Eos	1	4	**5**	7	5
EQs	0	3	4	6	0
ILFs	0	7	10	15	0
EIFs	0	5	**7**	10	0
Count Total					9

FP questions:

1. Does the system require backup and recovery? 0
2. Are the specialised data communications required to transfer information to or from the application? 0
3. Are there distributed processing functions? 0
4. Is performance critical? 0
5. Will the system run in an existing, heavily utilised operational environment? 0
6. Does the system require on-line data entry? 0
7. Does the on-line data entry require the input transaction to be built over multiple screens or operations? 0
8. Are the ILFs updated on-line? 0
9. Are the inputs, outputs, files, or inquiries complex? 2
10. Is the internal processing complex? 3
11. Is the code designed to be reusable? 4
12. Are conversion and installation included in the design? 0
13. Is the system designed for multiple installations in different organisations? 0
14. Is the application designed to facilitate change and for ease of use by the user? 1

FP = 9 x [0.65 + 0.01 x 10]
FP = 6.75
After finding both MCM and FP values Code Quality (CQ) should be measured as:

CQ = FP / MCM = 6.75 / 34171 = 0.000197536

To make this value more understandable, we use the equation 3.1

Accordingly the Code Quality of C++ program is given by;

= (6.75 / 34171) * 10000 => 1.97536

5 Conclusion

We have presented a multi-paradigm complexity metric. It is formulated by including most of the factors responsible for increasing the complexity of procedural and Object-Oriented code. Furthermore, code complexity is estimated by involving function point measurement which makes our proposal more realistic. For demonstrating our metric we have applied it on a case study. We aim to apply our metric to more projects to check it practical applicability. Application of our multi paradigm metric on other multi paradigms languages e.g. Java, and Python is also the task of future work.

References

1. Pfleeger, S.L., Atlee, J.M.: Software Engineering – Theory and Practice, 3rd International edn. Prentice-Hall, Englewood Cliffs (2006)
2. Sommerville, I.: Software Engineering, 7th edn. Addison-Wesley, Reading (2004)
3. Software Technology Support Centre Software Estimation, Measurement, and Metrics, http://www.stsc.hill.af.mil/resources/tech_docs/gsam3/chap13.pdf (last accessed 11.03.2011)
4. Basili, V.R.: Qualitative Software Complexity Models: A Summary. In: Tutorial on Models and Methods for Software Management and Engineering. IEEE Computer Society Press, Los Alamitos (1980)
5. Kearney, J.K., Sedlmeyer, R.L., Thompson, W.B., Gray, M.A., Adler, M.A.: Software Complexity Measurement, vol. 28, pp. 1044–1050 (1986)
6. Marco, L.: Measuring Software Complexity, http://cispom.boisestate.edu/cis320emaxson/metrics.htm (last accessed. 23.02.2011)
7. Misra, S.: Cognitive Complexity Measures: An Analysis' Modern Software Engineering Concepts and Practices: Advanced Approaches, pp. 263–279. IGI Global, USA (2011)
8. McCabe, T.J., Watson, A.H.: Software Complexity, McCabe and Associates, Inc., http://www.stsc.hill.af.mil/crosstalk/1994/12/xt94d12b.asp (last accessed 17.03.2011)
9. Metrics 1.3.6, http://metrics.sourceforge.net/ (last accessed 23.02.2011)
10. Misra, S., Akman, I.: A Model for Measuring Cognitive Complexity of Software, pp. 879–886. Springer, Heidelberg (2008)
11. Misra, S., Akman, I.: Unified Complexity Metric: A measure of Complexity. Proc. of National Academy of Sciences Section A. 80(2), 167–176 (2010)
12. TIOBE Software – The Coding Standards Company. Programming Community Index for February 2010. (last accessed 18.02.2011)
13. Da-wei, E.: The Software Complexity Model and Metrics for Object-Oriented. In: Proc IEEE Int. Workshop on Anti-counterfeiting, Security and Identification, pp. 464–469 (2007)
14. http://homepages.fh-regensburg.de/~mpool/mpool10/welcome.html

15. Westbrook, D.S.: A Multi-paradigm Language Approach to Teaching Principles of Programming Languages. In: Proc. 29th ASE/IEEE Frontiers in Education Conference, pp. 3–14 (1999)
16. Van Roy, P.: Department of Computing Science and Engineering, Catholic University of Louvain, `http://www.info.ucl.ac.be/~pvr/cvvanroy.html` (last accessed 17.03.2010)
17. Roger, S.P.: Software Engineering – A practitioner's approach, 6th edn. McGraw-Hill, New York (2005)
18. Misra, S., Cafer, F.: Estimation Complexity of programs in Python Languages. Technical Gazette 18(1), 1–10 (2011)
19. McCabe, T.J.: A Complexity Measure. IEEE Transactions Software Engineering 2(6), 308–320 (1976)
20. Halstead, M.H.: Elements of Software Science. Elsevier North-Holland, New York (1997)
21. Wang, Y., Shao, J.: A New Measure of Software Complexity Based on Cognitive Weights. Can. J. Elec. Computer Engineering, 69–74 (2003)
22. Costagliola, G., Tortora, G.: Class points:An approach for the size Estimation of Object-oriented systems. IEEE Transactions on Software Engineering 31(1), 52–74 (2005)
23. Misra, S.: Ibrahim Akman 'Weighted Class Complexity: A Measure of Complexity for Object Oriented Systems'. Journal of Information Science and Engineering 24, 1689–1708 (2008)
24. Chidamber, S.R., Kermer, C.F.: A Metric Suite for object oriented design. IEEE Transactions Software Engineering SE-6, 476–493 (1994)
25. Misra, S., Akman, I.: A new complexity metric based on cognitive informatics. In: Wang, G., Li, T., Grzymala-Busse, J.W., Miao, D., Skowron, A., Yao, Y. (eds.) RSKT 2008. LNCS (LNAI), vol. 5009, pp. 620–627. Springer, Heidelberg (2008)
26. Porkoláb, Z., Sillye, Á.: Towards a multiparadigm complexity measure. In: QAOOSE Workshop, ECOOP, Glasgow, pp. 134–142 (2005)
27. Norbert, P., Adam, S., Zoltan, P.: orkolab: Measuring the Complexity of Aspect-Oriented Programs with Multiparadigm Metric. In: Proc. of ECOOP 2006 Doctoral Symposium and PhD Students Workshop (2006), `http://www.ecoop.org/phdoos/ecoop2006ds/ws/pataki.pdf`

Appendix

Example Code in C

```
#include <iostream>
#include <string>
using namespace std;
// Colour
class Colour{
        void stars(int limit);
public:
        static char c;
        void getColour();
};
void Colour::getColour(){
        if (c=='s')
                cout<<"Yellow"<<endl;
```

```
        else if (c=='c')
                cout<<"Violet"<<endl;
        else if (c=='r')
                cout<<"Red"<<endl;
        else if (c=='o')
                cout<<"Orange"<<endl;
        else
                cout<<"White"<<endl;
        stars(5);
}
void Colour::stars(int limit){
        int outer_loop, inner_loop;
        for (outer_loop=limit; outer_loop>0; outer_loop--){
                for (inner_loop=1; inner_loop<=outer_loop; inner_loop++)
                        printf("*");
                printf("\n");
        }
}
// -----------
char Colour::c;

class Shapes {
public:
        Shapes(int px, int py):x(px),y(py) {}
        int x, y; //position
        virtual string type() = 0;
        virtual void info() {
                cout << endl << "figure: " << type() << endl;
                cout << "position: x=" << x << ", y=" << y << endl;
        }
};

class Figure1P : public Shapes {
public:
        Figure1P(int px, int py, int r):p1(r),Shapes(px, py) {}
        int p1;
        virtual void info() {
                Shapes::info();
                cout << "property 1: p=" << p1 << endl;
        }
};

class Square : public Figure1P {
public:
        Colour *its_colour;
        Square(int px, int py, int r):Figure1P(px, py, r) {}
        virtual string type() {
```

```
                Colour::c='s';
                its_colour->getColour();
                return "square";
        }
};

class Circle : public Figure1P {
public:
        Colour *its_colour;
        Circle(int px, int py, int r):Figure1P(px, py, r) {}
        virtual string type() {
                Colour::c='c';
                its_colour->getColour();
                return "circle";
        }
};

class Figure2P : public Figure1P {
public:
        Figure2P(int px, int py, int w, int h):p2(h),Figure1P(px, py, w) {}
        int p2;
        virtual void info() {
                Figure1P::info();
                cout << "property 2: p=" << p2 << endl;
        }
};

class Rectangle : public Figure2P {
public:
        Colour *its_colour;
        Rectangle(int px, int py, int w, int h):Figure2P(px, py, w, h) {}
        virtual string type() {
                Colour::c='r';
                its_colour->getColour();
                return "rectangle";
        }
};

class Oval : public Figure2P {
public:
        Colour *its_colour;
        Oval(int px, int py, int w, int h):Figure2P(px, py, w, h) {}
        virtual string type() {

                Colour::c='o';
                its_colour->getColour();
                return "oval";
        }
};
```

```
// Freeing memory
   void freeRAM(Shapes *objs[], int i){

         delete objs[i];

   }
   // --------------

   int main(void) {
         Shapes **objs = new Shapes*[35];
         // creating objects
         objs[0] = new Circle(7, 6, 55);
         objs[1] = new Rectangle(12, 54, 21, 14);
         objs[2] = new Square(19, 32, 10);
         objs[3] = new Oval(43, 10, 4, 3);
         objs[4] = new Square(3, 41, 3);
         bool flag=false;
         do {
                 cout << endl << "We have 5 objects with numbers 0..4" << endl;
                 cout << "Enter object number to view information about it " <<
endl;
                 cout << "Enter any other number to quit " << endl;
                 char onum; // in fact, this is a character, not a number
                 // this allows user to enter letter and quit...
                 cin >> onum;
                 // flag -- user have entered number 0..4
                 flag = ((onum >= '0')&&(onum <= '4'));
                 if (flag)
                         objs[onum-'0']->info();
         } while(flag);
         for(int i=0;i<5;i++)
                 freeRAM(objs,i);
         delete [] objs;      return(0);
   }
```

Flow Sensitive-Insensitive Pointer Analysis Based Memory Safety for Multithreaded Programs

Mohamed A. El-Zawawy

Department of Mathematics
Faculty of Science
Cairo University
Giza 12613
Egypt
maelzawawy@cu.edu.eg

Abstract. The competency of pointer analysis is crucial for many compiler optimizations, transformations, and checks like memory safety. The potential interaction between threads in multithreaded programs complicates their pointer analysis and memory-safety check. The trade-off between accuracy and scalability remains a main issue when studying these analyses. In this work, we present novel approaches for the pointer analysis and memory safety of multithreaded programs as simply structured type systems.

In order to balance accuracy and scalability, the type system proposed for pointer analysis of multithreaded programs is flow-sensitive and it invokes another flow-insensitive type system for parallel constructs. Therefore the proposed pointer analysis is described as flow sensitive-insensitive. The third type system presented in this paper takes care of memory safety of multithreaded programs and is an extension of the type system of pointer analysis. Programs having types in memory-safety type system, are guaranteed to be memory safe. Type derivations serve as proofs for correctness of the result of every pointer analysis and memory-safety check. Such proofs are required in the area of proof-carrying code.

Keywords: Pointer analysis, memory safety, operational semantics, multithreaded programs, type systems.

1 Introduction

Two facts contribute to the enormous importance that the pointer analysis of multithreaded programs enjoys. One fact is that pointer information is important for many compiler optimizations and corrections [23]. The other fact is the growing interest in multithreading as a mainstream practice of programming. One important use of pointer information is to statically judge the memory safety of programs. That is to reason about (a) the existence of pointer arithmetics when they are not allowed by the syntax of the language and (b) the existence of dangling pointers (de-referencing of variables that contain no pointers). The fact that pointer analysis is a main tool in code parallelization magnifies importance of pointer analysis.

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 355–369, 2011.

For multithreaded programs, compilation and program analyses are challenging problems [17,23] because the potential interaction between various threads creates difficulty in extending techniques of compiling and analyzing sequential programs to cover multithreaded ones. For the pointer analysis, the interaction happens when a thread writes a pointer variable that is simultaneously accessed by another thread. Such interactions result in enlarging sets of pointers that variables may point to. For the sake of correctness, analyses of multithreaded programs must conveniently interpret the interaction between various threads.

Pointer analysis [7,9] calculates information about memory locations that are pointed to by program pointers. The memory safety analysis aims at statically proves that the program does not treat pointers illegally according to the language syntax. One way to classify techniques of pointer analysis and memory safety is according to flow-sensitivity of approaches i.e. into flow-sensitive and flow-insensitive. The approaches of flow-insensitive neglect the program's flow of control. Hence in these techniques the program statements are dealt with as if they are executable any number of times in any probable order. Flow-insensitive approaches [1,2] are believed to be less precise and more efficient than flow sensitive ones. Therefore there is a trade-off between using flow-sensitive and flow-insensitive approaches. In this paper, we introduce pointer analysis and memory safety techniques for multithreaded programs. Aiming at capturing the advantages of the two approaches, the proposed techniques mix flow sensitivity and insensitivity.

Typically, static analysis of programs is done in an algorithmic style through which algorithms work on control-flow graphs of programs rather than on their syntactic structures. The algorithmic style has the drawback of working like a black box in the sense that it is not clear how the analysis was done. Therefore in applications like proof-carrying code or certified code, where a proof for the correctness of analysis results is required to be delivered with the result, the algorithmic style is not an ideal choice. Type systems have established themselves as good tools to carry static analysis instead of algorithms specially for applications of proof-carrying code [3,14,20]. Type rules are relatively easy to interpret and type derivations are good format for required proofs. The techniques presented in this paper for pointer analysis and memory safety of multithreaded programs are in the form of type systems.

Figure 1 presents the programming language that we study. The language is the simple *while* language [11] enriched with commands for structured parallel constructs and pointer manipulations. Join-fork constructs, conditionally spawned threads, and parallel loops are parallel constructs included in the language. The join-fork (*par*) construct begins the execution of its threads concurrently at the beginning of the construct and

$$\begin{aligned}
& n \in \mathbb{Z},\ x \in \mathit{Var},\ \mathit{and} \oplus \in \{+,-,\times\}\\
e \in \mathit{Aexprs} &::= x \mid n \mid e_1 \oplus e_2\\
b \in \mathit{Bexprs} &::= \mathit{true} \mid \mathit{false} \mid \neg b \mid e_1 = e_2 \mid e_1 \leq e_2 \mid b_1 \wedge b_2 \mid b_1 \vee b_2\\
S \in \mathit{Stmts} &::= x := e \mid x := \&y \mid *x := e \mid x := *y \mid \mathit{skip} \mid S_1; S_2 \mid \mathit{if}\ b\ \mathit{then}\ S_t\ \mathit{else}\ S_f \mid\\
& \quad\ \mathit{while}\ b\ \mathit{do}\ S_t \mid \mathit{par}\{\{S_1\},\ldots,\{S_n\}\} \mid \mathit{par\text{-}if}\{(b_1,S_1),\ldots,(b_n,S_n)\} \mid \mathit{par\text{-}for}\{S\}.
\end{aligned}$$

Fig. 1. The programming language

then waits for the accomplishment of these executions at the end of the *par* construct. The parallel loop construct, *par-for*, executes in parallel a statically unknown number of threads that have the same code (the loop body). The construct including conditionally spawned threads is that of *par-if* and it executes in parallel its n threads where the execution of thread (b_i, S_i) includes the execution of S_i only if b_i is true.

This paper presents a new technique for pointer analysis of multithreaded programs. Type systems are basic tools of our proposed technique which mixes concepts of flow sensitive and insensitive analyses. More precisely, the technique has the form of a type system that is flow-sensitive and that invokes another type system which is flow-insensitive when parallel constructs are encountered. The paper also presents another type system that checks memory-safety of multithreaded programs and that utilizes the pointer information obtained by our pointer analysis. We show that if a program has types in the memory safety type system then it is memory-safe in the sense that it is guaranteed not to abort due to illegal pointer operations.

Motivation

```
1.  x := &y;
2.   *x := 2;
3.  par{
4.      {y := &x}
5.      {y := 5;
6.       *x := &z}
7.      };
8.  x := &z;
9.  z := 2;
10.  y := *z
```

Program point	Pointer information
first point	$\{t \mapsto \emptyset \mid t \in Var\}$
between lines 1 & 2	$\{x \mapsto \{y\}, t \mapsto \emptyset \mid x \neq t\}$
points between 2 & 8	$\{x \mapsto \{y\}, y \mapsto \{x, z\}, t \mapsto \emptyset \mid t \notin \{x, y\}\}$
points between 8 & 10	$\{x \mapsto \{z\}, y \mapsto \{x, z\}, t \mapsto \emptyset \mid t \notin \{x, y\}\}$
last point	$\{x \mapsto \{z\}, t \mapsto \emptyset \mid x \neq t\}$

Fig. 2. A motivating example together with its pointer analysis

The program in Figure 2 is a motivating example of the work presented in this paper. We note that this program aborts because the command at the last line de-references the variable z that has no address. Therefore this program is not memory safe. It is the task of this paper to introduce a technique that statically (without running programs) tests memory safety of multithreaded programs like this one. The information that z has no addresses can be inferred from the result of a pointer analysis for the program. Hence the paper introduces an efficient flow sensitive-insensitive pointer analysis whose results for our example program are in the table of Figure 2.

Contributions

Contributions of this paper are the following:

1. A new flow sensitive-insensitive pointer analysis technique for multithreaded programs.
2. An original type system for flow-insensitive pointer analysis of multithreaded programs.
3. An original analysis that checks memory safety of multithreaded programs.

Organization

The rest of the paper is organized in sections that present the following topics in the following order:

1. Related work.
2. The proposed technique (type systems) for pointer analysis of multithreaded programs.
3. The proposed technique (type system) for memory safety of multithreaded programs.
4. Operational semantics of our language (Figure 1).
5. Conclusion.

2 Related Work

Pointer Analysis and Memory Safety

The pointer analysis and memory safety for sequential programs have been studied extensively for decades [7,10,9,25,6]. Flow-sensitive pointer analyses [8,26,30], which are more natural to most applications, consider the order of program commands. Mostly these analyses perform an abstract interpretation of program using dataflow analysis to associate each program point with a points-to relation. Flow-insensitive pointer analyses [1,2] do not consider the order of program commands. Typically the output of these analyses, which are performed using a constraint-based approach, is a points-to relation that is valid all over the program. Clearly the flow-sensitive approach is more precise but less efficient than the flow-insensitive one. The work [25] presents a type and effect analysis for detecting memory and type errors in C source code.

Although problems of pointer analysis and memory safety for sequential programs were studied extensively, a little effort was done towards a pointer and a memory-safety analyses for multithreaded programs. In [22], a flow sensitive analysis for multithreaded programs was introduced. This analysis associates each program point with a triple of points-to relations. This in turn complicates the the analysis and creates a sort of redundancy in the collected points-to information. The work in [7] presents a design for a formal low-level multi-threaded language enriched with region-based memory management and synchronization constructs. Well-typed programs of this language are claimed to be memory safe and race free. Investigating the details of these approaches and our work makes it apparent that our work is simpler and more accurate than these approaches. Moreover our approach provides a proof for the correctness of the pointer analysis and memory safety for each program. To the best of our knowledge, such proof is not known to be provided by any other existing approach.

Type Systems in Program Analysis

The work in [3,14,20,6,19] is among the closest work to ours in the sense that it uses type systems to achieve the program analysis in a way similar to ours. The work in [14] shows that a good deal of program analysis can be done using type systems. More precisely, it proves that for every analysis in a certain class of data-flow analyses, there exists a type system such that a program checks with a type if and only if the type is a supertype for the set resulting from running the analysis on the program. The type system in [18] and the flow-logic work in [20], which is used in [19] to study security of the coordinated systems, are very similar to [14]. For the simple while language, the work in [3] introduces type systems for constant folding and dead code elimination and also logically proves correctness of optimizations. Safety policies for information flow, carrying-code abstraction, and resource usage were also casted using type systems [4,5]. Earlier, related work (with structurally-complex type systems) is [21].

To the best of our knowledge, our approach is the first attempt to use type systems to check memory safety based pointer analysis for multithreaded programs and associates every individual check with a justification for correctness.

Analysis of Multithreaded Programs

The analysis of multithreaded programs is a challenging area [23] that receives growing interest; threading complicates the program analysis. There are several directions of research in this area [15,13]. Deadlock which results from round waiting to gain resources is a problem of multithreading computing. Researchers have developed various techniques for deadlock detection [12,27,28]. The aim of the analysis of synchronization constructs [24,29] is to explore how the synchronization actions apart executions of program segments. The result of this analysis can be used by compiler to appropriately add join-fork constructs. Data race describes the situation when a memory location is accessed by two threads (one of them writes in the location) without synchronization. One direction of research focuses on data race detection [16].

The problem with almost all the work refereed to above is that it does not apply to pointer programs. More precisely, for some of the work the application is possible only if we have the result of a pointer analysis for the input pointer program. The techniques presented in this paper have the advantage of being simpler and more reliable than the techniques refereed to above that would work in the presence of a pointer analysis.

3 Pointer Analysis

This section presents a new technique for pointer analysis of multithreaded programs that allow shared pointers to be updated simultaneously. Our technique is basically a flow-sensitive analysis that invokes flow-insensitive one for the analysis of parallel constructs; join-fork constructs, parallel loops, and conditionally spawned threads. Therefore the technique can be described as flow sensitive-insensitive. Both of the analyses take the form of compositional type systems that are simply structured. The pointer information calculated by the analyses have the form of types assigned to expressions and statements. The correctness of collected pointer information is proved by type derivations. Hence the presented analyses proceed by assigning a type to each program point of a statement (program). A points-to type associates each variable in the program with a conservative approximation of the addresses that may get into the variable. The soundness of all type systems presented in this paper is proved using the operational semantics presented in Section 5. The set of states of this semantics is denoted by Γ.

The following definition introduces the set of points-to types PTS and the relation $\models \subseteq \Gamma \times PTS$:

Definition 1. *1.* $Addrs = \{x' \mid x \in Var\}$.
2. $PTS = \{pts \mid pts : Var \to 2^{Addrs}\}$.
3. $pts \leq pts' \overset{\text{def}}{\iff} \forall x \in Var.\ pts(x) \subseteq pts'(x)$.
4. $\gamma \models pts \overset{\text{def}}{\iff} (\forall x \in Var.\ \gamma(x) \in Addrs \implies \gamma(x) \in pts(x))$.

We start with introducing a type system for flow-insensitive pointer analysis of multithreaded programs. Then a type system for flow-sensitive analysis that invokes the flow-insensitive type system is introduced.

3.1 Flow-Insensitive Pointer Analysis

The types, PTS, of our type system for flow-insensitive pointer analysis are introduced in the previous definition. The inference rules of the type system are the following:

$$\frac{}{x : pts, pts(x)} \quad \frac{}{n : pts, \emptyset} \quad \frac{}{e_1 \oplus e_2 : pts, \emptyset} \quad \frac{}{skip : pts} \quad \frac{e : pts, A \quad A \subseteq pts(x)}{x := e : pts}\ (:=^p_{insen})$$

$$\frac{y' \in pts(x)}{x := \&y : pts}\ (:= \&^p_{insen}) \quad \frac{\forall z' \in pts(x).\ z := e : pts}{*x := e : pts}\ (* :=^p_{insen})$$

$$\frac{\forall z' \in pts(y).\ x := z : pts}{x := *y : pts}\ (:= *^p_{insen}) \quad \frac{\forall i.\ S_i : pts}{par\{\{S_1\}, \ldots, \{S_n\}\} : pts}\ (par^p_{insen})$$

$$\frac{\forall i.\ S_i : pts}{par\text{-}if\{(b_1, S_1), \ldots, (b_n, S_n)\} : pts}\ (par\text{-}if^p_{insen}) \quad \frac{S : pts}{par\text{-}for\{S\} : pts}\ (par\text{-}for^p_{insen})$$

$$\frac{S_1 : pts \quad S_2 : pts}{S_1; S_2 : pts}\ (seq^p_{insen}) \quad \frac{S_t : pts \quad S_f : pts}{if\ b\ then\ S_t\ else\ S_f : pts}\ (if^p_{insen})$$

$$\frac{S_t : pts}{while\ b\ do\ S_t : pts}\ (whl^p_{insen}) \quad \frac{S : pts \quad pts \leq pts'}{S : pts'}\ (csq^p_{insen})$$

The judgements of an expression e and a statement S have forms $e : pts, A$ and $S : pts$, respectively. As formalized by Lemma 1, A is the collection of addresses that e may evaluate to in a state of type pts. The judgement of S guarantees that if the execution of S in a state of type pts terminates in a state γ', then γ' has type pts.

The inference rule $(:=^p_{insen})$ says that for the assignment statement to be of type pts, the set $pts(x)$ has to cover the set of addresses A. The other inference rules corresponding to other assignment commands are clarified similarly. A type invariant is necessary to type statements like *while*, *par*, *par-if*, and *par-for*. A fix-point algorithm can be used to find type invariants. The monotonicity of rules of the type system and the fact that the set of points-to types PTS is a complete lattice guarantee the convergence of the algorithm.

Lemma 1. *1. Suppose $e : pts, A$ and $\gamma \models pts$. Then $[\![e]\!]\gamma \in Addrs$ implies $[\![e]\!]\gamma \in A$.*
2. $pts \leq pts' \iff (\forall \gamma.\ \gamma \models pts \implies \gamma \models pts')$.

Proof. (1) is obvious. The right-to-left direction of (2) is proved as follows. Suppose $y' \in pts(x)$. Then the state $\{(x, y'), (t, 0) \mid t \in Var \setminus \{x\}\}$ is of type pts and hence of type pts' implying that $y' \in pts'(x)$. Therefore $pts(x) \subseteq pts'(x)$. Since x is arbitrary, $pts \leq pts'$. The other direction is easy.

Theorem 1. (*Soundness*) *Suppose that $S : pts$, $S : \gamma \rightsquigarrow \gamma'$, and $\gamma \models pts$. Then $\gamma' \models pts$.*

Proof. The proof is by structure induction on the type derivation. We demonstrate some cases.

- The case of $(:=^p_{insen})$: in this case $\gamma' = \gamma[x \mapsto [\![e]\!]\gamma]$. Therefore by the previous lemma $\gamma \models pts$ implies $\gamma' \models pts$.
- The case of $(* :=^p_{insen})$: in this case there exists $z \in Var$ such that $\gamma(x) = z'$ and $z := e : \gamma \rightsquigarrow \gamma'$. Because $\gamma \models pts$, $z' \in pts(x)$ and hence by assumption $z := e : pts$. Therefore by soundness of $(:=^p_{insen})$, $\gamma' \models pts$.
- The case of (par^p_{insen}): in this case there exist a permutation $\theta : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$ and $n + 1$ states $\gamma = \gamma_1, \ldots, \gamma_{n+1} = \gamma'$ such that for every $1 \leq i \leq n$, $S_{\theta(i)} : \gamma_i \rightarrow \gamma_{i+1}$. Also $\gamma_1 \models pts$. Therefore by the induction hypothesis $\gamma_2 \models pts$. Again by the induction hypothesis we get $\gamma_3 \models pts$. Therefore by a simple induction on n, we can show that $\gamma' = \gamma_{n+1} \models pts$.
- The case of $(par\text{-}for^p_{insen})$: in this case there exists n such that $par\{\overbrace{\{S\}, \ldots, \{S\}}^{n-times}\} : \gamma \rightsquigarrow \gamma'$. By induction hypothesis we have $S : pts$. By (par^p_{insen}) we conclude that $par\{\overbrace{\{S\}, \ldots, \{S\}}^{n-times}\} : pts$. Therefore by the soundness of (par^p_{insen}), $\gamma' \models pts'$.

3.2 Flow Sensitive-Insensitive Pointer Analysis

This section presents the basic type system that carries the pointer analysis of multi-threaded programs. For parallel constructs the type system calls the flow-insensitive type system presented in the previous subsection. Therefore the type system presented here is described as flow sensitive-insensitive analysis. The following are the rules of the type system:

$$\frac{}{n : pts \rightarrow \emptyset} \quad \frac{}{x : pts \rightarrow pts(x)} \quad \frac{}{e_1 \oplus e_2 : pts \rightarrow \emptyset} \quad \frac{e : pts \rightarrow A}{x := e : pts \rightarrow pts[x \mapsto A]}\ (:=^p_{sen})$$

$$\frac{}{x := \&y : pts \rightarrow pts[x \mapsto \{y'\}]}\ (:= \&^p_{sen}) \qquad \frac{}{skip : pts \rightarrow pts}$$

$$\frac{\forall z' \in pts(y).\ x := z : pts \rightarrow pts'}{x := *y : pts \rightarrow pts'}\ (:= *^p_{sen}) \quad \frac{\forall z' \in pts(x).\ z := e : pts \rightarrow pts'}{*x := e : pts \rightarrow pts'}\ (* :=^p_{sen})$$

$$\frac{par\{\{S_1\},\ldots,\{S_n\}\} : pts}{par\{\{S_1\},\ldots,\{S_n\}\} : pts \to pts}\,(par^p) \qquad \frac{S_1 : pts \to pts'' \quad S_2 : pts'' \to pts'}{S_1; S_2 : pts \to pts'}\,(seq^p_{sen})$$

$$\frac{par\text{-}if\{(b_1, S_1),\ldots,(b_n, S_n)\} : pts}{par\text{-}if\{(b_1, S_1),\ldots,(b_n, S_n)\} : pts \to pts}\,(par\text{-}if^p_{sen})$$

$$\frac{par\text{-}for\{S\} : pts}{par\text{-}for\{S\} : pts \to pts}\,(par\text{-}for^p_{sen}) \qquad \frac{S_t : pts \to pts' \quad S_f : pts \to pts'}{if\ b\ then\ S_t\ else\ S_f : pts \to pts'}\,(if^p_{sen})$$

$$\frac{S_t : pts \to pts}{while\ b\ do\ S_t : pts \to pts}\,(whl^p_{sen}) \qquad \frac{pts'_1 \leq pts_1 \quad S : pts_1 \to pts_2 \quad pts_2 \leq pts'_2}{S : pts'_1 \to pts'_2}\,(csq^p_{sen})$$

The judgements of an expression e and a statement S have forms $e : pts \to A$ and $S : pts \to pts'$, respectively. The intuition of these judgements are similar to that described in the previous section for corresponding judgments. A typical pointer analysis for a program S takes the form of a post-type derivation starting with the bottom type (mapping variables to $\emptyset$) as the pre-type.

Lemma 2. *Suppose $e : pts \to A$ and $\gamma \models pts$. Then $[\![e]\!]\gamma \in Addrs$ implies $[\![e]\!]\gamma \in A$.*

Theorem 2. (*Soundness*) *Suppose that $S : pts \to pts'$, $S : \gamma \rightsquigarrow \gamma'$, and $\gamma \models pts$. Then $\gamma' \models pts'$.*

Proof. The proof is by structure induction on the type derivation. Some cases are shown below.

- The case of $(:=^p_{sen})$: in this case $pts' = pts[x \mapsto A]$ and $\gamma' = \gamma[x \mapsto [\![e]\!]\gamma]$. Therefore by the previous lemma $\gamma \models pts$ implies $\gamma' \models pts'$.
- The case of $(* :=^p_{sen})$: in this case there exists $z \in Var$ such that $\gamma(x) = z'$ and $z := e : \gamma \rightsquigarrow \gamma'$. Because $\gamma \models pts$, $z' \in pts(x)$ and hence by assumption $z := e : pts \to pts'$. Therefore by soundness of $(:=^p_{sen})$, $\gamma' \models pts'$.
- The cases of (par^p_{sen}), $(par\text{-}for^p_{sen})$, and $(par\text{-}if^p_{sen})$ follow directly from soundness of the type system of the previous subsection (Theorem 1).

4 Memory Safety

This section presents a new technique that statically studies the memory safety of multithreaded programs. A memory safe program is one that is guaranteed not to attempt any illegal operations with pointers like dereferencing a variable that has no pointer. The proposed technique is a type system that extends the type system of pointer analysis presented in the previous section. The extension takes the form of another type component added to points-to types. The new component is meant to capture for each program point the set of variables that must contain addresses. The resulting types of the memory-safety type system can be seen as a refitment of the points-to types. This is reflected by the fact that the pointer information collected by the pointer analysis is used in the rules of memory-safety type system.

The following definition introduces the set of safety-types MS and the relation $\models \subseteq \Gamma \times MS$:

Definition 2. – *A safety type is a pair of a points-to type pts and a subset $v \subseteq Var$.*
- $(pts, v) \leq (pts', v') \overset{\text{def}}{\Longleftrightarrow} pts \leq pts'$ *and* $v \supseteq v'$.
- *A state γ has type (pts, v), denoted by $\gamma \models (pts, v)$, if $\gamma \models pts$ and $\forall x \in v.\ \gamma(x) \in Addrs$.*

Inference rules of our type system for memory safety are the following:

$$\frac{y \in v}{y : (x, pts, v) \curvearrowright v \cup \{x\}}\,(y_1^m) \qquad \frac{y \notin v}{y : (x, pts, v) \curvearrowright v \setminus \{x\}}\,(y_2^m) \qquad \frac{}{n : (x, pts, v) \curvearrowright v \setminus \{x\}}\,(n^m)$$

$$\frac{\forall y \in FV(e_1 \oplus e_2).\ pts(y) = \emptyset}{e_1 \oplus e_2 : (x, pts, v) \curvearrowright v \setminus \{x\}}\,(\oplus^m) \qquad \frac{x := e : pts \to pts' \quad e : (x, pts, v) \curvearrowright v'}{x := e : (pts, v) \curvearrowright (pts', v')}\,(:=^m)$$

$$\frac{}{skip : (pts, v) \curvearrowright (pts, v)} \qquad \frac{x \in v \quad \forall z' \in pts(x)(z := e : (pts, v) \curvearrowright (pts', v'))}{*x := e : (pts, v) \curvearrowright (pts', v')}\,(* :=^m)$$

$$\frac{y \in v \quad \forall z' \in pts(y)(x := z : (pts, v) \curvearrowright (pts', v'))}{x := *y : (pts, v) \curvearrowright (pts', v')}\,(:= *^m)$$

$$\frac{x := \&y : pts \to pts'}{x := \&y : (pts, v) \curvearrowright (pts', v \cup \{x\})}\,(:= \&^m) \qquad \frac{S : (pts, v \cap v') \curvearrowright (pts, v')}{par\text{-}for\{S\} : (pts, v) \curvearrowright (pts, v')}\,(par\text{-}for^m)$$

$$\frac{S_i : (pts, v \cap \cap_{j \neq i} v_j) \curvearrowright (pts, v_i)}{par\{\{S_1\}, \ldots, \{S_n\}\} : (pts, v) \curvearrowright (pts, \cap_i v_i)}\,(par^m) \qquad \frac{S_t : (pts, v) \curvearrowright (pts, v)}{while\ b\ do\ S_t : (pts, v) \curvearrowright (pts, v)}\,(whl^m)$$

$$\frac{S_1 : (pts, v) \curvearrowright (pts'', v'') \quad S_2 : (pts'', v'') \curvearrowright (pts', v')}{S_1; S_2 : (pts, v) \curvearrowright (pts', v')}\,(seq^m)$$

$$\frac{par\{\{if\ b_1\ then\ S_1\ else\ skip\}, \ldots, \{if\ b_n\ then\ S_n\ else\ skip\}\} : (pts, v) \curvearrowright (pts, v')}{par\text{-}if\{(b_1, S_1), \ldots, (b_n, S_n)\} : (pts, v) \curvearrowright (pts, v')}\,(par\text{-}if^m)$$

$$\frac{S_t : (pts, v) \curvearrowright (pts', v') \quad S_f : (pts, v) \curvearrowright (pts', v')}{if\ b\ then\ S_t\ else\ S_f : (pts, v) \curvearrowright (pts', v')}\,(if^m)$$

$$\frac{(pts_1', v_1') \leq (pts_1, v_1) \quad S : (pts_1, v_1) \curvearrowright (pts_2, v_2) \quad (pts_2, v_2) \leq (pts_2', v_2')}{S : (pts_1', v_1') \curvearrowright (pts_2', v_2')}\,(csq^m)$$

The judgement of an expression e has the form $e : (x, pts, v) \curvearrowright v'$. We note that for a type (pts, v) and a variable x the post type v' does not always exist. Therefore for a type (pts, v) and a variable x such a judgement does not always exist. When exists the judgement guarantees that the statement $x := e$ does not abort when executed in a state of type (pts, v) and that if the execution terminates in a state γ', then v' contains variables that contain addresses according to γ'. The judgement of a statement S has the form $S : (pts, v) \curvearrowright (pts', v')$. For a given statement S and a pre-type (pts, v) such a judgement does not always exist. This judgement, if exists, guarantees that S does not abort at any state of type (pts, v) and if the execution of S in a state of type (pts, v) terminates in a state γ', then γ' has type (pts', v'). A typical memory safety analysis for a program S takes the form of a post-type derivation starting with the bottom type (where pts maps variables to $\emptyset$ and $v = \emptyset$) as the pre-type.

The inference rule $(\oplus^m)$ reflects that in order for the assignment $x := e_1 \oplus e_2$ to succeed both of the operands e_1 and e_2 have to be guaranteed not to be pointers. By Lemma 1 this is guaranteed if $\forall y \in FV(e_1 \oplus e_2)(pts(y) = \emptyset)$. In this case x is assigned a number therefore gets removed from v to produce v'. In the rules $(* :=^m)$ and $(:= *^m)$ the variables x and y, respectively, have to belong to v to guarantee that the dereferenced variables x and y do contain addresses before execution. For the rule (par^m), according to semantics of the join-fork command, *par*, one possibility is that the execution of a specific thread S_i starts before the execution of any other thread starts. Another possibility is that the execution starts after executions of all other threads end. Of course there are many other possibilities in between. Consequently, the analysis of the thread S_i must consider all such possibilities. This is reflected in the pre-type of S_i and the post-type of the *par* command. Similar explanations clarify the rules $(par\text{-}if^m)$ and $(par\text{-}for^m)$.

We note that a type invariant is required to type a *while* statement. Also to achieve the analysis for one of the *par*'s threads we need to know the analysis results for all other threads. However obtaining these results requires the result of analyzing the first thread. Therefore there is a kind of circularity in rule (par^m). Similar situations are in rules $(par\text{-}if^m)$ and $(par\text{-}for^m)$. Such issues can be treated using a fix-point algorithm. The convergence of this algorithm is guaranteed as the rules of our type system are monotone and the set of points-to types *PTS* is a complete lattice.

Lemma 3. *1.* $(pts, v) \le (pts', v') \Longrightarrow (\forall \gamma \in \Gamma.\ \gamma \models (pts, v) \Longrightarrow \gamma \models (pts', v'))$.

2. Suppose $e : (x, pts, v) \curvearrowright v'$ *and* $\gamma \models (pts, v)$. *Then*

(a) $[\![e]\!] \neq !$, *and*

(b) If $x := e : \gamma \to \gamma'$, *then* $\forall y \in Var.\ (\gamma'(y) \in Addrs \Longrightarrow y \in v')$.

Proof. The proof of the first item is easy. The proof of the second item is by induction on the structure of type derivation:

- The case of the rule (y_1^m): in this case $\gamma' = \gamma[x \mapsto \gamma(y)]$ and $v' = v \cup \{x\}$. Since $y \in v$, $\gamma(y) \in Addrs$. Therefore $\gamma'(x) \in Addrs$ which justifies adding x to v.
- The case of the rule (y_2^m): in this case $\gamma' = \gamma[x \mapsto \gamma(y)]$ and $v' = v \setminus \{x\}$. Since $y \notin v$, $\gamma(y)$ is not guaranteed to be an address. Consequently $\gamma'(x)$ is not guaranteed to be an address which justifies removing x from v.
- The case of the rule $(\oplus^m)$: in this case $\gamma' = \gamma[x \mapsto [\![e_1 \oplus e_2]\!]\gamma]$ and $v' = v \setminus \{x\}$. $[\![e_1 \oplus e_2]\!] \in \mathbb{Z}$ because $\forall y \in FV(e_1 \oplus e_2)(pts(y) = \emptyset)$, which guarantees that $\forall y \in FV(e_1 \oplus e_2).\ \gamma(y) \in \mathbb{Z}$. Consequently $\gamma'(x) \in \mathbb{Z}$ which justifies removing x from v.

Theorem 3. *(soundness and memory safety) Suppose* $S : (pts, v) \curvearrowright (pts', v')$ *and* $\gamma \models (pts, v)$. *Then*

1. *S does not abort at γ i.e.* $S : \gamma \not\rightsquigarrow abort$.
2. *If* $S : \gamma \rightsquigarrow \gamma'$ *then* $\gamma' \models (pts', v')$.

Proof. The proof is by structure induction on the type derivation. Some cases are shown as follows:

– The case of $(:=^m)$: this case results from the previous lemma and the soundness of pointer analysis.
– The case of $(* :=^m)$: in this case there exists $z \in Var$ such that $\gamma(x) = z'$ because $x \in v$. We have $z' \in pts(x)$ because $\gamma \models pts$. By induction hypothesis $z := e$ does not abort at γ and consequently neither does $*x := e$. For (2) we have $z := e : \gamma \rightsquigarrow \gamma'$. By assumption we have, $z := e : (pts, v) \curvearrowright (pts', v')$. Therefore by soundness of $(:=^m)$, $\gamma' \models (pts', v')$.
– The case of (par^m): (1) We outline the proof that executing the n threads in any order starting from a state γ of type (pts, v) does not abort. Suppose that $\theta : \{1, \ldots, n\} \to \{1, \ldots, n\}$ is a permutation. $\gamma \models (pts, v)$ implies $\gamma \models (pts, v \cap \cap_{j \neq \theta(1)} v_j)$. Therefore by induction hypothesis $S_{\theta(1)}$ does not abort at γ. Then either $S_{\theta(1)}$ enters an infinite loop at γ or the execution terminates at a state γ_2 which is by induction hypothesis of type $(pts, v_{\theta(1)})$. Therefore γ_2 is of type $(pts, v \cap \cap_{j \neq \theta(2)} v_j)$. Hence a simple induction on n shows (1).
(2) In this case there exist a permutation $\theta : \{1, \ldots, n\} \to \{1, \ldots, n\}$ and $n+1$ states $\gamma = \gamma_1, \ldots, \gamma_{n+1} = \gamma'$ such that for every $1 \leq i \leq n,\ S_{\theta(i)} : \gamma_i \to \gamma_{i+1}$. Also $\gamma_1 \models (pts, v)$ implies $\gamma_1 \models (pts, v \cap \cap_{j \neq \theta(1)} v_j)$. Therefore by the induction hypothesis $\gamma_2 \models (pts, v_{\theta(1)})$. This implies $\gamma_2 \models (pts, v \cap \cap_{j \neq \theta(2)} v_j)$. Again by the induction hypothesis we get $\gamma_3 \models (pts, v_{\theta(2)})$. Therefore by a simple induction on n, we can show that $\gamma' = \gamma_{n+1} \models (pts, v_{\theta(n)})$ which implies $\gamma' \models (pts', v') = (pts, \cap_j v_j)$.
– The case of $(par\text{-}for^m)$: (1) Similarly to the previous case it is easy to show that the *par-for* command does not abort at any state of type (pts, v).
(2) In this case there exists n such that $par\{\overbrace{\{S\}, \ldots, \{S\}}^{n-times}\} : \gamma \rightsquigarrow \gamma'$. By induction hypothesis we have $S : (pts, v \cap v') \curvearrowright (pts, v')$. By (par^m) we conclude that $par\{\overbrace{\{S\}, \ldots, \{S\}}^{n-times}\} : (pts, v) \curvearrowright (pts, v')$. Therefore by the soundness of (par^m), $\gamma' \models (pts', v')$.

5 Operational Semantics

This section presents an operational semantics for constructs of the language (Figure 1) we study. The semantics of the *par* command can be approximately interpreted as if the threads are executed sequentially in an arbitrary order. By definitions, the semantics of *par-for* and *par-if* are expressed using that of the *par* construct. Introducing operation semantics is one way to describe the meanings of the constructs of our programming language, including the parallel constructs. This is equivalent to defining a transition relation $\rightsquigarrow$ between states which are defined as follows.

Definition 3. *1. $Addrs = \{x' \mid x \in Var\}$ and $Val = \mathbb{Z} \cup Addrs$.*
2. A state is either an abort or a map $\gamma \in \Gamma = Var \longrightarrow Val$.

Rather than that arithmetic and Boolean operations are not allowed on pointers, the semantics of arithmetic and Boolean expressions are defined as usual.

$$[\![n]\!]\gamma = n \quad [\![\&x]\!]\gamma = x' \quad [\![x]\!]\gamma = \gamma(x) \quad [\![true]\!]\gamma = true \quad [\![false]\!]\gamma = false$$

$$[\![*x]\!]\gamma = \begin{cases} \gamma(y) & \text{if } \gamma(x) = y', \\ ! & \text{otherwise.} \end{cases} \qquad [\![e_1 \oplus e_2]\!]\gamma = \begin{cases} [\![e_1]\!]\gamma \oplus [\![e_2]\!]\gamma & \text{if } [\![e_1]\!]\gamma, [\![e_2]\!]\gamma \in \mathbb{Z}, \\ ! & \text{otherwise.} \end{cases}$$

$$[\![\neg A]\!]\gamma = \begin{cases} \neg([\![A]\!]\gamma) & \text{if } [\![A]\!]\gamma \in \{\mathit{true}, \mathit{false}\}, \\ ! & \text{otherwise.} \end{cases} \qquad [\![e_1 = e_2]\!]\gamma = \begin{cases} ! & \text{if } [\![e_1]\!]\gamma = ! \text{ or } [\![e_2]\!]\gamma = !, \\ \mathit{true} & \text{if } [\![e_1]\!]\gamma = [\![e_2]\!]\gamma \neq !, \\ \mathit{false} & \text{otherwise.} \end{cases}$$

$$[\![e_1 \leq e_2]\!]\gamma = \begin{cases} ! & \text{if } [\![e_1]\!]\gamma \notin \mathbb{Z} \text{ or } [\![e_2]\!]\gamma \notin \mathbb{Z}, \\ [\![e_1]\!]\gamma \leq [\![e_2]\!]\gamma & \text{otherwise.} \end{cases}$$

$$\text{For } \diamond \in \{\wedge, \vee\},\ [\![b_1 \diamond b_2]\!]\gamma = \begin{cases} ! & \text{if } [\![b_1]\!]\gamma = ! \text{ or } [\![b_2]\!]\gamma = !, \\ [\![b_1]\!]\gamma \diamond [\![b_2]\!]\gamma & \text{otherwise.} \end{cases}$$

The inference rules of our semantics (transition relation) are defined as follows:

$$\frac{[\![e]\!]\gamma = !}{x := e : \gamma \rightsquigarrow \mathit{abort}} \qquad \frac{[\![e]\!]\gamma \neq !}{x := e : \gamma \rightsquigarrow \gamma[x \mapsto [\![e]\!]\gamma]} \qquad \frac{\gamma(x) = z' \quad z := e : \gamma \rightsquigarrow \mathit{state}}{*x := e : \gamma \rightsquigarrow \mathit{state}}$$

$$\frac{\gamma(x) \notin \mathit{Addrs}}{*x := e : \gamma \rightsquigarrow \mathit{abort}} \qquad \frac{}{x := \&y : \gamma \rightsquigarrow \gamma[x \mapsto y']} \qquad \frac{\gamma(y) = z' \quad x := z : \gamma \rightsquigarrow \gamma'}{x := *y : \gamma \rightsquigarrow \gamma'}$$

$$\frac{\gamma(y) \notin \mathit{Addrs}}{x := *y : \gamma \rightsquigarrow \mathit{abort}} \qquad \frac{}{\mathit{skip} : \gamma \rightsquigarrow \gamma} \qquad \frac{S_1 : \gamma \rightsquigarrow \mathit{abort}}{S_1; S_2 : \gamma \rightsquigarrow \mathit{abort}} \qquad \frac{S_1 : \gamma \rightsquigarrow \gamma'' \quad S_2 : \gamma'' \rightsquigarrow \mathit{state}}{S_1; S_2 : \gamma \rightsquigarrow \mathit{state}}$$

$$\frac{[\![b]\!]\gamma = !}{\mathit{if}\ b\ \mathit{then}\ S_t\ \mathit{else}\ S_f : \gamma \rightsquigarrow \mathit{abort}} \qquad \frac{[\![b]\!]\gamma = \mathit{true} \quad S_t : \gamma \rightsquigarrow \mathit{state}}{\mathit{if}\ b\ \mathit{then}\ S_t\ \mathit{else}\ S_f : \gamma \rightsquigarrow \mathit{state}}$$

$$\frac{[\![b]\!]\gamma = \mathit{false} \quad S_f : \gamma \rightsquigarrow \mathit{state}}{\mathit{if}\ b\ \mathit{then}\ S_t\ \mathit{else}\ S_f : \gamma \rightsquigarrow \mathit{state}} \qquad \frac{[\![b]\!]\gamma = !}{\mathit{while}\ b\ \mathit{do}\ S_t : \gamma \rightsquigarrow \mathit{abort}} \qquad \frac{[\![b]\!]\gamma = \mathit{false}}{\mathit{while}\ b\ \mathit{do}\ S_t : \gamma \rightsquigarrow \gamma}$$

$$\frac{[\![b]\!]\gamma = \mathit{true} \quad S : \gamma \rightsquigarrow \gamma'' \quad \mathit{while}\ b\ \mathit{do}\ S_t : \gamma'' \rightsquigarrow \mathit{state}}{\mathit{while}\ b\ \mathit{do}\ S_t : \gamma \rightsquigarrow \mathit{state}} \qquad \frac{[\![b]\!]\gamma = \mathit{true} \quad S : \gamma \rightsquigarrow \mathit{abort}}{\mathit{while}\ b\ \mathit{do}\ S_t : \gamma \rightsquigarrow \mathit{abort}}$$

- **Join-Fork:**

$$\frac{}{\mathit{par}\{\{S_1\}, \ldots, \{S_n\}\} : \gamma \rightsquigarrow \gamma'}\ \dagger \qquad \frac{}{\mathit{par}\{\{S_1\}, \ldots, \{S_n\}\} : \gamma \rightsquigarrow \mathit{abort}}\ \ddagger$$

  † there exist a permutation $\theta : \{1, \ldots, n\} \to \{1, \ldots, n\}$ and $n + 1$ states $\gamma = \gamma_1, \ldots, \gamma_{n+1} = \gamma'$ such that for every $1 \leq i \leq n$, $S_{\theta(i)} : \gamma_i \to \gamma_{i+1}$.
  ‡ there exist m such that $1 \leq m \leq n$, a one-to-one map $\beta : \{1, \ldots, m\} \to \{1, \ldots, n\}$, and $m + 1$ states $\gamma = \gamma_1, \ldots, \gamma_{m+1} = \mathit{abort}$ such that for every $1 \leq i \leq m$, $S_{\beta(i)} : \gamma_i \to \gamma_{i+1}$.
- **Conditionally Spawned Threads:**

$$\frac{\mathit{par}\{\{\mathit{if}\ b_1\ \mathit{then}\ S_1\ \mathit{else}\ \mathit{skip}\}, \ldots, \{\mathit{if}\ b_n\ \mathit{then}\ S_n\ \mathit{else}\ \mathit{skip}\}\} : \gamma \rightsquigarrow \gamma'}{\mathit{par\text{-}if}\{(b_1, S_1), \ldots, (b_n, S_n)\} : \gamma \rightsquigarrow \gamma'}$$

$$\frac{\mathit{par}\{\{\mathit{if}\ b_1\ \mathit{then}\ S_1\ \mathit{else}\ \mathit{skip}\}, \ldots, \{\mathit{if}\ b_n\ \mathit{then}\ S_n\ \mathit{else}\ \mathit{skip}\}\} : \gamma \rightsquigarrow \mathit{abort}}{\mathit{par\text{-}if}\{(b_1, S_1), \ldots, (b_n, S_n)\} : \gamma \rightsquigarrow \mathit{abort}}$$

- **Parallel Loops:**

$$\frac{\exists n.\, par\{\overbrace{\{S\},\ldots,\{S\}}^{n-times}\} : \gamma \rightsquigarrow \gamma'}{par\text{-}for\{S\} : \gamma \rightsquigarrow \gamma'} \qquad \frac{\exists n.\, par\{\overbrace{\{S\},\ldots,\{S\}}^{n-times}\} : \gamma \rightsquigarrow abort}{par\text{-}for\{S\} : \gamma \rightsquigarrow abort}$$

Some comments on the inference rules are in order. The execution of assignment command (:=) aborts at a state only if the semantics of the expression e at that state includes arithmetics on pointers. The semantics of the indirect assignment commands ($* :=$ and $:= *$) uses that of direct assignment (:=) and has one more source of abortion which happens due to de-referencing that is unsafe. One source for abortion of *if* and *while* statements is the un-computability of Boolean conditions of these statements which happens when a Boolean operation is tried on pointers. The execution of *par* command, the main parallel command, amounts to starting at the begin of construct the concurrent execution of all the command threads and waiting at the end of the construct for the termination of these executions. This is approximated by the inference rules for *par* command. The semantics of the other parallel commands (*par-if* and *par-for*) is defined by means of the inference rules for *par* command.

6 Conclusion

Many compiler optimizations, transformations, and checks like memory safety are directly affected by the efficiency of the crucial program analysis of pointer analysis. One factor that complicates the pointer analysis and memory-safety check of multithreaded programs is the potential interaction between threads. A main issue when studying these analyses for programs is the trade-off between accuracy and scalability. Novel approaches, in the form of simply structured type systems, for the pointer analysis and memory safety of multithreaded programs are presented in this paper.

For the sake of balancing accuracy and scalability, a flow-insensitive type system for parallel constructs is invoked by the main flow-sensitive type system proposed for pointer analysis of multithreaded programs. Hence the proposed technique is classified as flow sensitive-insensitive. This paper extends the proposed type system for pointer analysis to introduce the third type system of the paper which takes care of memory safety of multithreaded programs. Memory safe is guaranteed for programs typed in the proposed memory-safety type system. In the proposed techniques, the result of every pointer analysis and memory-safety check is associated with a correctness proof which has the form of a type derivation. These proofs are necessary in many applications like proof-carrying code (certified code).

References

1. Adams, S., Ball, T., Das, M., Lerner, S., Rajamani, S.K., Seigle, M., Weimer, W.: Speeding up dataflow analysis using flow-insensitive pointer analysis. In: Hermenegildo, M.V., Puebla, G. (eds.) SAS 2002. LNCS, vol. 2477, p. 230. Springer, Heidelberg (2002)
2. Anderson, P., Binkley, D., Rosay, G., Teitelbaum, T.: Flow insensitive points-to sets. Information & Software Technology 44(13), 743–754 (2002)

3. Benton, N.: Simple relational correctness proofs for static analyses and program transformations. In: Jones, N.D., Leroy, X. (eds.) POPL, pp. 14–25. ACM, New York (2004)
4. Beringer, L., Hofmann, M., Momigliano, A., Shkaravska, O.: Automatic certification of heap consumption. In: Baader, F., Voronkov, A. (eds.) LPAR 2004. LNCS (LNAI), vol. 3452, pp. 347–362. Springer, Heidelberg (2005)
5. Besson, F., Jensen, T.P., Pichardie, D.: Proof-carrying code from certified abstract interpretation and fixpoint compression. Theor. Comput. Sci. 364(3), 273–291 (2006)
6. El-Zawawy, M.A.: Program optimization based pointer analysis and live stack-heap analysis. International Journal of Computer Science Issues 8(2), 98–107 (2011)
7. Gerakios, P., Papaspyrou, N., Sagonas, K.F.: Race-free and memory-safe multithreading: design and implementation in cyclone. In: Kennedy, A., Benton, N. (eds.) TLDI, pp. 15–26. ACM, New York (2010)
8. Hardekopf, B., Lin, C.: Semi-sparse flow-sensitive pointer analysis. In: Shao, Z., Pierce, B.C. (eds.) POPL, pp. 226–238. ACM, New York (2009)
9. Hind, M.: Pointer analysis: haven't we solved this problem yet? In: PASTE, pp. 54–61 (2001)
10. Hind, M., Pioli, A.: Which pointer analysis should i use? In: ISSTA, pp. 113–123 (2000)
11. Hoare, C.: An axiomatic basis for computer programming. Commun. ACM 12(10), 576–580 (1969)
12. Kim, B.-C., Jun, S.-W., Hwang, D.J., Jun, Y.-K.: Visualizing potential deadlocks in multithreaded programs. In: Malyshkin, V. (ed.) PaCT 2009. LNCS, vol. 5698, pp. 321–330. Springer, Heidelberg (2009)
13. Knoop, J., Steffen, B.: Code motion for explicitly parallel programs. In: PPOPP, pp. 13–24 (1999)
14. Laud, P., Uustalu, T., Vene, V.: Type systems equivalent to data-flow analyses for imperative languages. Theor. Comput. Sci. 364(3), 292–310 (2006)
15. Lee, J., Midkiff, S.P., Padua, D.A.: Concurrent static single assignment form and constant propagation for explicitly parallel programs. In: Huang, C.-H., Sadayappan, P., Sehr, D. (eds.) LCPC 1997. LNCS, vol. 1366, pp. 114–130. Springer, Heidelberg (1998)
16. Leung, K.-Y., Huang, Z., Huang, Q., Werstein, P.: Maotai 2.0: Data race prevention in view-oriented parallel programming. In: PDCAT, pp. 263–271. IEEE Computer Society, Los Alamitos (2009)
17. Midkiff, S.P., Padua, D.A.: Issues in the optimization of parallel programs. In: Padua, D.A. (ed.) ICPP (2), pp. 105–113. Pennsylvania State University Press (1990)
18. Naik, M., Palsberg, J.: A type system equivalent to a model checker. ACM Trans. Program. Lang. Syst. 30(5), 1–24 (2008)
19. Nicola, R.D., Gorla, D., Hansen, R.R., Nielson, F., Nielson, H.R., Probst, C.W., Pugliese, R.: From flow logic to static type systems for coordination languages. Sci. Comput. Program. 75(6), 376–397 (2010)
20. Riis Nielson, H., Nielson, F.: Flow logic: A multi-paradigmatic approach to static analysis. In: Mogensen, T.Æ., Schmidt, D.A., Sudborough, I.H. (eds.) The Essence of Computation. LNCS, vol. 2566, pp. 223–244. Springer, Heidelberg (2002)
21. Palsberg, J., O'Keefe, P.: A type system equivalent to flow analysis. ACM Trans. Program. Lang. Syst. 17(4), 576–599 (1995)
22. Rugina, R., Rinard, M.C.: Pointer analysis for structured parallel programs. ACM Trans. Program. Lang. Syst. 25(1), 70–116 (2003)
23. Sarkar, V.: Challenges in code optimization of parallel programs. In: de Moor, O., Schwartzbach, M.I. (eds.) CC 2009. LNCS, vol. 5501, pp. 1–1. Springer, Heidelberg (2009)
24. Tian, C., Nagarajan, V., Gupta, R., Tallam, S.: Automated dynamic detection of busy-wait synchronizations. Softw., Pract. Exper. 39(11), 947–972 (2009)
25. Tlili, S., Debbabi, M.: Interprocedural and flow-sensitive type analysis for memory and type safety of c code. J. Autom. Reasoning 42(2-4), 265–300 (2009)

26. Wang, J., Ma, X., Dong, W., Xu, H.-F., Liu, W.: Demand-driven memory leak detection based on flow- and context-sensitive pointer analysis. J. Comput. Sci. Technol. 24(2), 347–356 (2009)
27. Wang, Y., Kelly, T., Kudlur, M., Lafortune, S., Mahlke, S.A.: Gadara: Dynamic deadlock avoidance for multithreaded programs. In: Draves, R., van Renesse, R. (eds.) OSDI, pp. 281–294. USENIX Association (2008)
28. Xiao, X., Lee, J.J.: A true o(1) parallel deadlock detection algorithm for single-unit resource systems and its hardware implementation. IEEE Trans. Parallel Distrib. Syst. 21(1), 4–19 (2010)
29. Xu, C., Che, Y., Fang, J., Wang, Z.: Optimizing adaptive synchronization in parallel simulators for large-scale parallel systems and applications. In: CIT, pp. 131–138. IEEE Computer Society, Los Alamitos (2010)
30. Yu, H., Xue, J., Huo, W., Feng, X., Zhang, Z.: Level by level: making flow- and context-sensitive pointer analysis scalable for millions of lines of code. In: Moshovos, A., Steffan, J.G., Hazelwood, K.M., Kaeli, D.R. (eds.) CGO, pp. 218–229. ACM, New York (2010)

HProcessTOOL: A Support Tool in the Harmonization of Multiple Reference Models

César Pardo[1], Francisco Pino[1], Félix García[2], Francisco Romero Romero[2], Mario Piattini[2], and Maria Teresa Baldassarre[3]

[1] IDIS Research Group
Electronic and Telecommunications Engineering Faculty,
University of Cauca, Calle 5 No. 4 - 70.
Kybele Consulting Colombia (Spinoff)
Popayán, Cauca, Colombia
{cpardo,fjpino}@unicauca.edu.co
[2] Alarcos Research Group
Institute of Information Technologies & Systems,
University of Castilla-La Mancha, Paseo de la Universidad 4, Ciudad Real, España
{Felix.Garcia,FranciscoA.Romero,Mario.Piattini}@uclm.es
[3] Department of Informatics, University of Bari
SER&Practices, SPINOFF, Via E. Orabona 4, 70126, Bari, Italy
baldassarre@di.uniba.it

Abstract. If companies are to fulfil their business goals then they must implement more than one software process improvement or information technology management model. The heterogeneity of these models signifies that their harmonization in accordance with company goals has become a key initiative. It is therefore necessary to provide companies with suitable software tools which facilitate the implementation and management of the activities, methods, techniques and reference models involved in a harmonization project, thus allowing the harmonization to be properly carried out. This paper therefore presents the HProcessTOOL which guides harmonization projects by supporting specific techniques, and supports their management by controlling and monitoring the resulting harmonization projects. The tool has been applied in two case studies, and has allowed the work products, effort, time and roles involved in the harmonization projects, and the knowledge generated, to be correctly managed.

Keywords: Multi-model, Multiple, Reference models, Harmonization, Software Process Improvement, (SPI), Harmonization strategy, Software Tool.

1 Introduction

Various improvement methodologies such as IDEAL [1], PDCA [2], PmCOMPETISOFT [3] and Agile SPI-Process [4] have been defined to support the process improvement of enterprises, These methodologies have taken multiple reference models such as ISO 27001, ISO 9001, CMMI and ISO 90003 as their basis. Other models, such as ITIL [5], COBIT [6], RISK IT [7] and VAL IT [8], have also been defined to

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 370–382, 2011.

support Information Technology Management. The implementation and institutionalization of these approaches have allowed companies to improve, mature, acquire and institutionalise best practices and management systems, such as: (i) Information Security Management Systems (ISMS), (ii) Information Technology Governance Processes (IT Governance), (iii) quality management systems, or even those in much more specific domains such as (iv) software development, (v) software maintenance and (vi) operation.

Given that enterprises currently need to resolve various business and operative needs independently of their work areas, they are obliged to use more than one reference model at the same time. However, since each model defines its own structure of process entities, definitions and quality systems [9], this issue increases the complexity involved in the implementation of multi-models in a single organization. It also implies greater efforts, time and associated costs than a conventional SPI environment (which is characterized by its use of only one reference model). This type of environment accordingly requires a specialized technical infrastructure (tools and methodologies), which allows the harmonization of multiple reference models to be supported.

In the last four years several efforts have been made to define proposals which attempt to provide methodologies that support the harmonization of multiple models, e.g. the PrIME project of the SEI [10], the ARMONÍAS project of the ALARCOS research group [11], or Enterprise SPICE [12], among others. However, although it is possible to see an increase in the interest in proposals defined for this type of environment, few efforts have been focused on providing technological support through software tools. It is therefore necessary to provide tools and a specialized technical infrastructure with which to both support the harmonization of multiple models and strengthen this research domain. A software tool would undoubtedly facilitate the guidance, implementation and management of any harmonization project.

Bearing all this in mind, this paper describes a WEB tool, denominated as HProcessTOOL, which allows the realization of an organization's harmonization projects to be managed, controlled and monitored. HProcessTOOL is a wizard that guides the step-by-step harmonization of multiple models. More precisely, it allows us to: configure a harmonization strategy; choose a widespread harmonization strategy (WHS); mange generated knowledge and store the information from any model through the use of a common schema or Common Structure of Process Entities (CSPE), making the information collected from the models available for future work. This tool supports the Harmonization Process, which is part of a Harmonization Framework (HF) developed within the ARMONÍAS project. The HF integrates a set of techniques, methods, recommendations and concepts defined to support the harmonization of multiple models independently of the implementation approach of the models involved. A summary of both the Harmonization Process and the Harmonization Framework is presented in [13].

The paper is organized as follows. Section 2 presents an overview of the related works. Section 3 is divided into 3 parts: the first sets out an overview of the Harmonization Process, the second describes the Architecture of HProcessTOOL and the third sets out the modules of HProcessTOOL which have been developed. Section 4 presents a summary of the case studies in which the tool has been applied. Finally, our conclusions and future work are outlined in Section 5.

2 Related Work

Several software tools that offer control and planning in projects can be used as a basis for managing projects. Although these tools provide functions to support project management, they do not provide features focused on the harmonization of multi-models.

We are not aware of any other attempts to address the harmonization of multiple models through a tool. Therefore, and since the harmonization of multiple models is closely related to process improvement, we believe that it is important to analyse the differences between other tools developed to support software process improvement and our proposal. This will make it possible to verify whether the aforementioned tools can be used in these kinds of environments or whether a special type of tool, such as HProcessTOOL, is necessary. Table 1 shows a search of related works in which those

Table 1. Tools for process improvement and harmonization of multiple models

Type of environment focused on		MKS Integrity Suit	SIMPLe	QMIM Quality Organizer	HProcessTOOL
Type of tool	Desktop	X	--NA--	--NA--	--NA--
	Web	--NA--	X	X	X
Type of license	Free	X	--NA--	?	--NA--
	Commercial	--NA--	X	?	X
Knowledge base or Supported models		CMM [14], CMMI [15].	CMMI [15]	CMMI and other models such as ISO standards, ISO-IEC standards and 9 Hungarian standards.	Any. However, CMMI, ISO 20000, ISO 27001, ITIL V3, COBIT 4.1, RISK IT, BASEL II, VAL IT and ISO 27002 are currently stored.
Technique, method, process or model taken as guideline		?	IDEAL [1]	It supports the QMIM Framework (Quality through Managed Improvement and measurement)	Harmonization Process [13], ARMONÍAS Project [11]. Also supports the homogenization and comparison technique (see [9] and [16] respectively).
Does it support process improvement?		X	X	It only allows a CMMI-based assessment to be carried out.	It can be used as a support tool for process improvement when multiple models are involved.
Does it support the harmonization of two or more models?: multiple models.		--NA--	--NA--	?	X
Type of support		Monitoring, Control of changes (applications and documented process) and design of process.	Supports the documentation of organizational processes and an improvement project in general. It allows the generation of automatic creation of activities.	Quality organizer to improve the understanding and relationships between models, using QMIM Framework as a basis.	Provides a wide solution through which to manage, control and monitor harmonization projects with multiple models. This tool allows repetitive actions to be carried out and reduces the cognitive load of the individuals involved in a project of this type.
Reference		[17]	[18]	[19]	[20]

Some Conventions Used→ NA: Not Applicable, X: Applicable, ?: No information found.

tools that do not offer features oriented towards process improvement and the harmonizing of multiple models are excluded. The literature retrieved from this analysis has been classified into two categories: (i) tools that support the improvement process and (ii) tools that support the harmonization of multiple models. Other features such as: type of tool, type of license or supported models have also been analysed.

Upon considering the situation set out in Table 1, it is possible to observe that MKS Integrity Suit and SIMPLe are not independent of the reference models supported. They use related guidelines to support software process improvement. They do not, therefore, use a technique, method, process or methodology to support the harmonization of multiple models. The aim of QMIM Quality Organizer is to facilitate the understanding and selection of the best model and/or standard from an assessment of the current situation in Hungarian software companies. Although it contains information about the most popular quality approaches, standards and models, it does not define methods, techniques or tools to support the comparison, integration or unification of multiple models. Moreover, it is only focused on the software domain.

In contrast to existing tools, HProcessTOOL offers support which allows the harmonization of multiple models to be managed independently of the reference model to be harmonized. It can also be used as a support tool to address software process improvement when multiple models are involved, and additionally permits the harmonization of models with regard to software.

3 HProcessTOOL: WEB Tool for Supporting Harmonization Projects

HProcessTOOL is a WEB tool that provides a solution with which to manage, control and monitor harmonization projects. This tool allows repetitive actions to be carried out and reduces the cognitive load of the individuals involved in a project in which multiple reference models are necessary. HProcessTOOL uses the Harmonization Process defined in the HF as a basis (section 2.1). The Harmonization Process is employed to guide a work team through any harmonization project in which it is necessary to harmonize multiple models. It also supports other elements defined in the Harmonization Framework such as providing the person responsible with a step-by-step wizard guide to the configuration of a harmonization strategy during the execution, (see 2.1). HProcessTOOL is also a flexible environment, since it is independent of the reference models to be harmonized, and therefore harmonizes the structure of the models involved through the use of the Common Structure of Process Entities (CSPE) defined in [9]. This signifies that all the models chosen by an organization will be harmonized at the level of their process entity, which simplifies the implementation of other techniques, such as comparison and integration.

A brief summary of the harmonization process that HProcessTOOL uses, HProcessTOOL's Architecture and its characteristics, is presented in the following sections.

3.1 The Harmonization Process

The main objective of the harmonization process (Figure 1) is to establish the harmonization strategy, which describes the techniques and activities that must be performed in order to harmonize multiple models [13]. The Harmonization Process consists of four main activities: (i) Starting, (ii) Analysis and Definition, (iii) Execution and (iv) Revision, depending on the harmonization strategy selected. The Harmonization Process allows one or more harmonization iterations to be configured in the execution activity. An iterative and incremental approach makes it easier to manage the complexity related to the harmonization strategy to be executed in this activity.

The process additionally describes a set of roles: Performer (P), Process Engineer (PE), person Responsible for Managing the Process of Harmonization (RMPH) and Steering Group (SG). It also describes five main work products: Analysis of Needs and Identification of Cases Prior to Harmonization (PT01_A2_ANICPH), Harmonization Strategy (PT01_A2_HS), Implementation Report of the Harmonization Strategy (PT02_A2_IRHS) and Knowledge Base (PT02_2_KBHC). A more detailed description of the Harmonization Process edited with the EPF Composer is presented in [11], and a detailed summary of its elements can be seen in [13].

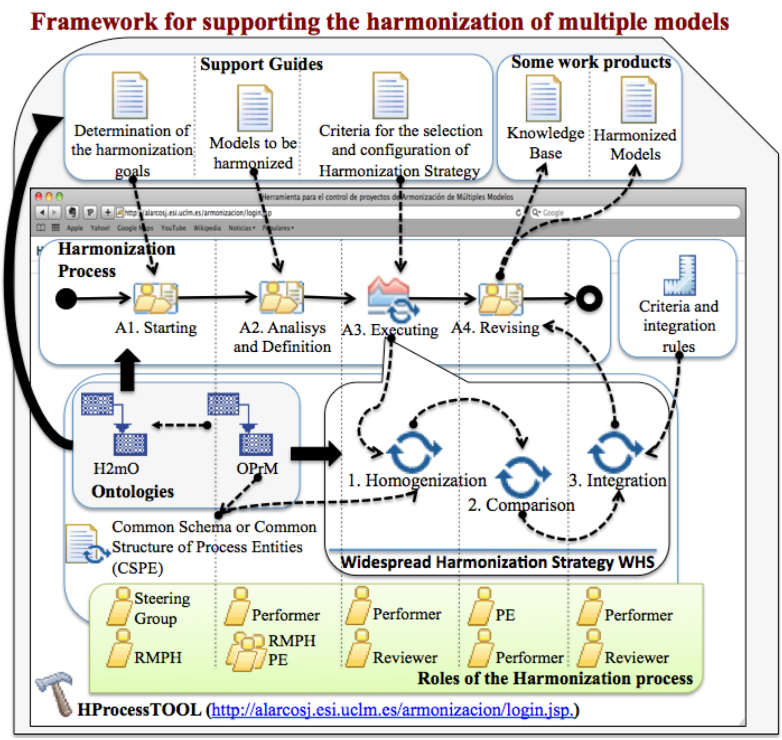


Fig. 1. Framework for supporting the harmonization of multiple models (taken from [21])

3.2 Architecture and Technological Considerations

HProcessTOOL is a flexible and scalable system which uses a client-server architecture that is applied by means of three layers, as follows:

- The *interface layer* (see left-hand side of Figure 2), also known as the presentation layer or application, contains the information and logic needed to submit requests and receive answers from the client. Its main components are the servlets and JavaServer Pages (JSP). Since our aim was to display rich analysis charts of a professional quality, we decided to use JFreeChart plug-in [22].
- The *control layer* (see right-hand side of Figure 2) contains the components (set of classes) of which the behavior and logic functionality core of HProcessTOOL is formed. This was developed by using Java JEE.
- The *data layer* (see right-hand side of Figure 2) is responsible for the persistence of the information generated by the Web tool. It is made up of a MySQL database.

Figure 2 shows a summary of the three layers of which HProcessTOOL is formed.

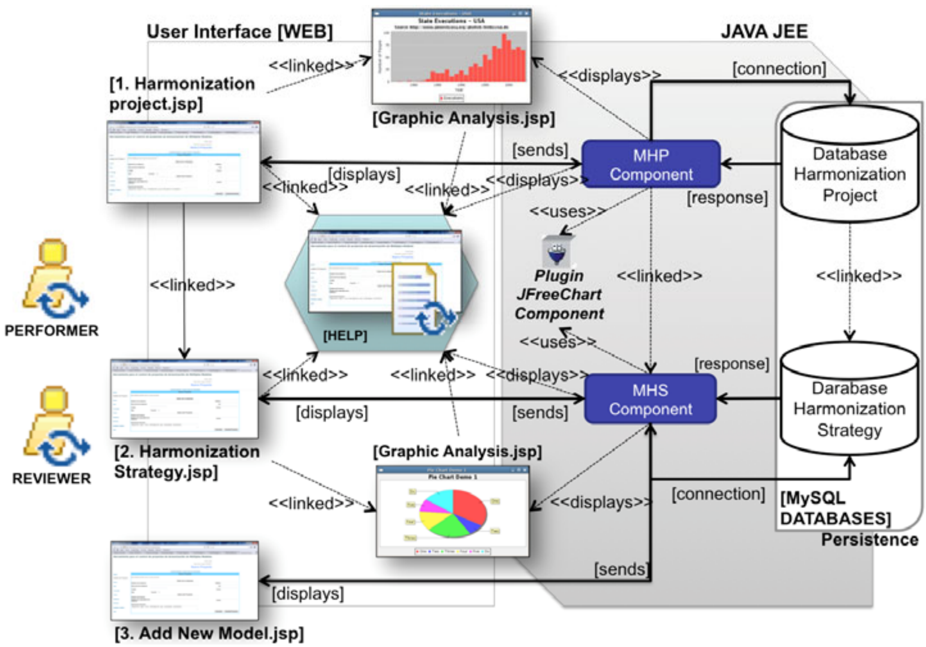


Fig. 2. General Architecture of HProcessTOOL

As regards the kind of users supported by the tool, HProcessTOOL supports two roles: the *Performer* which is based directly on the *Performer (P)* role defined in the Harmonization Process and the *Reviewer*, which is an adaption of the *Reviewer (R)* role defined in the mapping technique presented in [16].

The P is the person who is responsible for managing the harmonization project, analysing models and implementing the harmonization techniques. The review of the tasks and work performed by the P is carried out by the R, who verifies the reliability of the results obtained in both the harmonization process and the harmonization strategy. Both the P and R therefore have access to all the modules developed in HProcessTOOL.

3.3 Modules of HProcessTOOL

HProcessTOOL supports five main functionalities, as is shown in the general use case diagram in Figure 3.

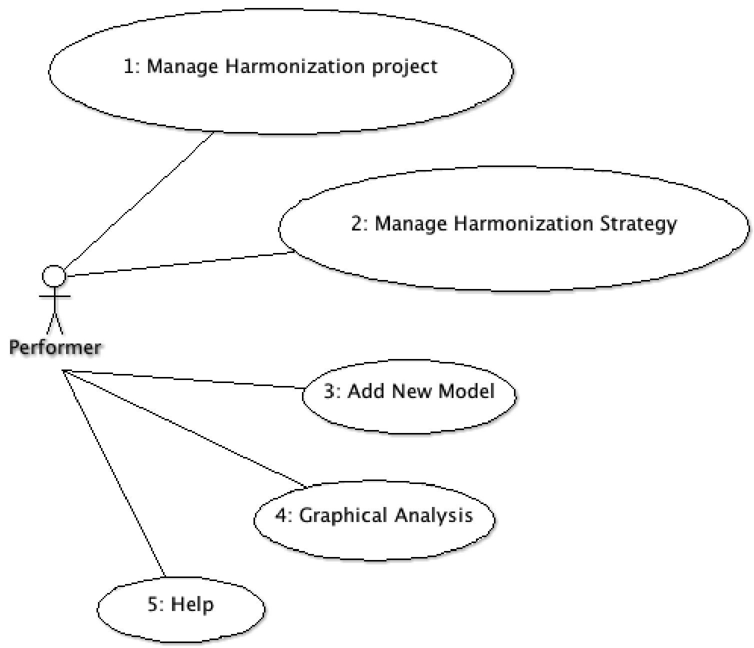


Fig. 3. Use case diagram of HProcessTOOL

As can be observed in Figure 3, the main functionalities supported are:

(1). *Management of the Harmonization Project (MHP).* This module allows a multiple model harmonization project to be managed, controlled and monitored. Tool's basis consists of the activities, tasks, steps, work products and certain roles such as the P role defined in the Harmonization Process (see Figure 1). The management carried out through this module is focused on four variables, which are:

- The control activity.
- Effort (estimated, actually made and number of people involved).
- The control date (starting and finishing date).
- Documents associated with the performance of each activity.

The control of the activities performed is carried out by using a check box which allows completed activities to be controlled. We have considered it important to add a field to store the additional information as part of the generation and management of a

Knowledge Base (KB). The KB can store issues such as: problems, positive aspects, negative aspects, solutions, notes, and so forth. The KB option is present in each activity, task and role. In general, the MHP uses a database, which allows all the information related to this module to be stored. Figure 6 shows an example of the management of a harmonization project.

(2). *Management of the Harmonization Strategy (MHS).* This module, which is also called MHS, allows the definition and systematic configuration of a harmonization strategy to be carried out. The MHS is divided into two options or sub-modules: either the configuration of a New Harmonization Strategy (NHS) or a Widespread Harmonization Strategy (WHS). These are described as follows:

- The NHS can be configured from a set of methods and/or techniques used by the organization. The tool has a specific module with which to create the activities, tasks and roles used to describe the process that is necessary.
- The WHS is a set of three stages or techniques:

  i). *Homogenization*, which provides the most suitable tools to harmonize the models involved through the addition of their information by means of a Common Schema or Common Structure of Process Entities (CSPE).

  ii). *Comparison*, which allows the identification of differences and similarities between a set of models to be carried out.

  iii). *Integration*, which provides the support needed to combine and/or unify the best practices of different models. A detailed summary of these techniques can be found in [9] and [16] respectively. The tool currently supports the homogenization and comparison technique. However, the integration technique is being developed from the results obtained from a case study whose main objective was the integration of a set of models to support the procedures and activities related to Information Technology (IT) Government in banking. A detailed analysis of the management of this is presented in Section 4, along with another case study.

Both NHS and WHS depend on a harmonization project having been created and opened. Figure 4 shows an example of the comparison between ISO 20000 and ISO/IEC 27001, which is performed through the harmonization strategy defined for Case Study 1 (explained in Section 4).

(3). *Add a new model.* This module allows the information in a new model to be added through a homogenization technique used by the WHS and the CSPE structure. However, unlike the MHS component, it does not depend on a project having been created and opened. Figure 5 shows Clause 4.2 of ISO 27001 which was added through the CSPE.

(4). *Graphical Analysis.* This module allows the data stored in MHP and MHS to be deployed through a graphical analysis by means of the JFreechart plug-in [22]. Some of the graphical analyses that can be carried out with the tool are: Gantt diagrams, activity, task and role diagrams in which, amongst other things, the time and effort that were estimated and were actually taken/made are analyzed.

Comparison of models ISO 27001 and ISO/IEC 20000

GP2: 4. Planning and Implementation of Service Management/ P1: Process 4.1 Planning of Service Management (Planning)/ A1: 1. Clause 4.1.1 defines the Service Management's Reach.

	T1: The service management's reach should be defined as part of the service management plan.	T2: The management should define the reach as part of its management responsibilities.
Degree of Relationship	P (1 de 2)	

ISO 27001/GPI: 4. Information Security Management System/P1: Clause 4.2 Creation and Management of ISMS/A1 Clause 4.2.1 refers to the creation of the ISMS.

T1: Define the reach and limits of the ISMS in terms of the characteristics of the business activity, those of the organization, its location, its activities and technology, including the details and justification of any exclusion from the reach.	75% [0.00-1.00] It has been founded a relationship	0.0% [0.00-1.00] There is any relationship
T2: Define an ISMS policy according to the characteristics of the business activity, those of the organization, its location, its activities and its technology.	0.0% [0.00-1.00] There is any relationship	0.0% [0.00-1.00] There is any relationship

Fig. 4. Comparison between ISO 20000 and ISO/IEC 27001

Introduce task

Homogenization

This interface allows the information about a Task to be introduced.

Name of model: ISO/IEC 27001
Process: Clause 4.2-Creation and Management of the ISMS

Activity: Clause 4.2.1 – Activity 4.2.1
Description: Refers to creation of the ISMS

Open Task

ID	Task	Description	
T1	Task 1	Define the ISMS's reach and limits in terms of the characteristics of the business activity, those of the organization, its location, its activities and technology, including the details of and justification for any exclusion from the reach.	○
T10	Task 10	Create a declaration of applicability, which should include: the control objects and the controls selected, along with the justification for their selection; the control objects and the controls currently used; and the exclusion of any control object and control from Annex A, along with the justification for this exclusion.	○
T2	Task 2	Define a policy for the ISMS in accordance with the characteristics of the business activity, those of the organization, its location, its activities and technology.	○
T3	Task 3	Define a focus for the organization's risk management.	○
T4	Task 4	Identify risks.	○
T5	Task 5	Analyze and evaluate risks.	○
			Open task

Fig. 5. Clause 4.2 of ISO 27001 stored through CSPE

(5). *Help.* This module offers two sorts of aids. The first offers a step-by-step guide to the harmonization process, which can be seen in the WEB in [11], and the second is a step-by-step guide as to how to use each of the functionalities of HProcess-TOOL. The latter can only be obtained through the WEB tool.

4 Applications of the Tool

The tool presented in this paper has been successfully used in the management of two different harmonization projects or case studies. These are described as follows:

1. The first harmonization project was focused on supporting the management of the harmonization of ISO 27001 and ISO 20000 part 2 as a requirement of a consultancy company called Audisec, which offers consultancy in these standards. The main objectives were to identify the differences and similarities between these standards, the level to which they are complementary, and to resolve the discrepancies between them, thus facilitating the certification of organizations in the ISO 20000 standard by considering the efforts made in previous certifications obtained for ISO 27001.
2. The second harmonization project was a research project, which sought the unification of a set of models to support different needs in Information Technology Governance when applicable to the Superintendence of Banks in Guatemala, and the banking sector in general. The model obtained integrates some of the processes of COBIT 4.1, VAL IT, RISK IT, ISO 27002 and ITIL V3, and the principles of Basel II, all of which go under the name of ITGSM. The integration process has initially been performed for the definition of 22 of the 44 processes proposed by ITGSM. A detailed summary of the ITGSM model is presented in [23].

A detailed summary of the management of both case studies is presented in [13] and [24] respectively. The management, control and monitoring were all performed by using HProcessTOOL. This allowed a suitable management of the work products, effort, time, roles involved and knowledge generated to be performed. It also allowed an appropriate and broad analysis of the projects to be made through the use of charts concerning the data stored from each project, Figure 6 shows an example of the management of the ITGSM harmonization project (we maintain the original screen shot, which is in Spanish).

The HProcessTOOL, through MHS, also allowed the configuration of a harmonization strategy according to the needs of each case, e.g. in the first case the MHS component allowed the techniques supported by the WHS component to be used, that is, it only included the homogenization and comparison techniques. However, since in the second case it was necessary to unifying multiple models, the tool did not support the integration of the models. This was therefore documented by using a word processor.

From the results obtained and the feedback received it is possible to state that companies that have not used HProcessTOOL could benefit from the Knowledge Base generated with regard to the models' information and relationships established in the comparisons.

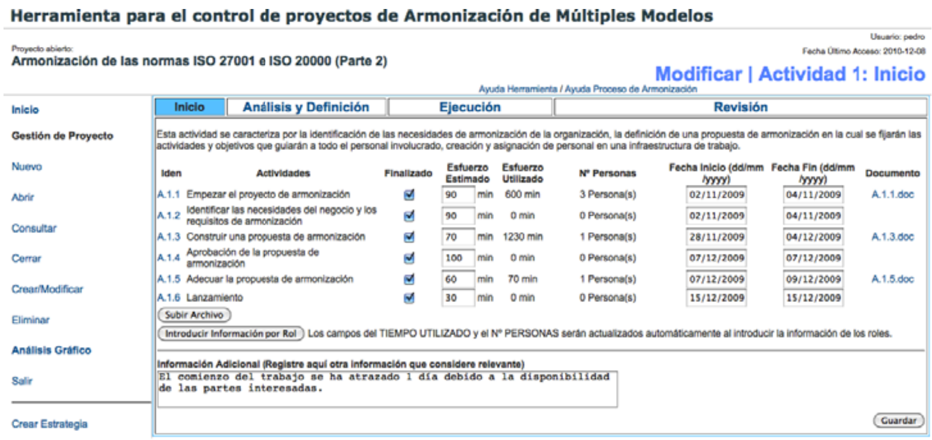


Fig. 6. Example of the management of the ITGSM harmonization project

5 Conclusions and Future Work

At present, and given the existence of different business and operative needs, enterprises are obliged to use more than one reference model at the same time. This kind of environment requires more time, effort and associated costs than the conventional environments for software process improvement, which are characterized by their use of only one model. This signifies that the tools and methodologies defined to support SPI projects are not applicable to multi-model environments. It is therefore necessary to provide tools and a specialized technical infrastructure with which to support the harmonization of multiple models and to strengthen this research domain.

This paper has presented a Web tool to support the harmonization of multiple models. HProcessTOOL provides a suitable work environment, independent of the reference models to be harmonized. It also allows the time, effort and resources used in a harmonization project to be optimized. HProcessTOOL has additionally been developed to provide support for any type of company, i.e., micro, small, medium and large enterprises. However, given that small companies have a single combination of features, it is possible that they will benefit most from our tool. We do not wish to infer that HProcessTOOL supports all the problems related to the harmonization of multiple models. However, the main objective of HProcessTOOL is to support the principal needs identified according to our experience.

Two harmonization projects have been managed with the first version of HProcessTOOL. The results obtained have been successful since the tool has allowed the following to be carried out: the identification and resolution of discrepancies between ISO 27001 and ISO 20000 part 2; support to be provided for the homogenization and comparison of BASEL II, COBIT 4.1, VAL IT, RISK IT, ISO 27002 and ITIL V3.

Since the harmonization project of the ITGSM Model was initially performed for the definition of 22 of the 44 processes proposed, as future work we aim to track this case study and to discover whether the tool has also allowed the 22 remaining processes to be successfully managed. We additionally wish to discover whether the use of HProcessTOOL has implied a reduction in the effort and costs associated with the implementation of harmonization projects without a software tool. In a similar vein, we also aim to include the module concerning the integration technique in the widespread harmonization technique and to improve the tool using the feedback received from new harmonization projects.

Acknowledgments. This work has been funded by the projects: ARMONÍAS (JCCM of Spain, PII2I09-0223-7948) and PEGASO/MAGO (MICINN and FEDER of Spain, TIN2009-13718-C02-01). Acknowledgements by Francisco J. Pino to the University of Cauca where he works as Associate Professor.

References

1. McFeeley, R.: IDEAL: A Users Guide for Software Process Improvement, Handbook CMU/SEI-96-HB-001. Technical report, Software Engineering Institute, Carnegie Mellon University (1996)
2. Deming, E.W.: Out of the Crisis. MIT Center for Advanced Engineering Study (1982)
3. Pino, F.J., Alegría, J.A.H., Vidal, J.C., García, F., Piattini, M.: A Process for Driving Process Improvement in VSEs. In: Proceedings of the International Conference on Software Process: Trustworthy Software Development Processes, pp. 342–353. Vancouver, B. C, Canada (2009)
4. Hurtado, J., Pino, F., Vidal, J., Pardo, C., Fernandez, L.: Agile SPI: Software Process Agile Improvement, A Colombia Approach to Software Process Improvement in Small Software Organizations. In: Piattini, M., Oktaba, H. (eds.) Software Process Improvement for Small and Medium Enterprises: Techniques and Case Studies, vol. 1, pp. 177–192. Idea Group Inc., USA (2008)
5. ITIL: Information Technology Infrastructure Library V3 (2010), `http://www.itil-officialsite.com/`
6. ITGI: COBIT 4.1: Framework, control objectives, management guidelines and maturity models. IT Governance Institute, EEUU (2007)
7. ITGI: Risk IT: Framework for Management of IT Related Business Risks. IT Governance Institute (2009), `http://www.isaca.org/`
8. ITGI: VAL IT Framework 2.0. IT Governance Institute, EEUU (2008)
9. Pardo, C., Pino, F., García, F., Piattini, M.: Homogenization of Models to Support multi-model processes in Improvement Environments. In: 4th International Conference on Software and Data Technologies, ICSOFT 2009, Sofía, pp. 151–156 (2009)
10. SEI: The PrIME Project (2010), `http://www.sei.cmu.edu/process/research/prime-details.cfm`
11. ARMONÍAS: A Process for Driving Multi-models Harmonization, ARMONÍAS Project (2009), `http://alarcos.esi.uclm.es/armonias/`
12. SPICE: Enterprise SPICE. An enterprise integrated standards-base model (2008), `http://www.enterprisespice.com/`
13. Pardo, C., Pino, F.J., García, F., Piattini, M., Baldassarre, M.T.: A Process for Driving the Harmonization of Models. In: The 11th International Conference on Product Focused Software Development and Process Improvement (PROFES 2010). Second Proceeding: Short Papers, Doctoral Symposium and Workshops, Markku Oivo, M.A.B., Matias Vierimaa, Limerick, pp. 51–54 (2010)

14. Paulk, M.C., Curtis, B., Chrissis, M., Weber, C.: Capability Maturity Model for Software, Version 1.1 (Technical Report CMU/SEI-93-TR-024). Technical report, Software Engineering Institute (1993)
15. SEI: CMMI for Development V1.2 Technical Report CMU/SEI-2006-TR-008. Technical report, Software Engineering Institute (2006)
16. Pino, F., Balssarre, M.T., Piattini, M., Visaggio, G.: Harmonizing maturity levels from CMMI-DEV and ISO/IEC 15504. Journal of Software Maintenance and Evolution: Research and Practice 22, 279–296 (2010)
17. MKS: MKS Integrity Suit. (2010), `http://www.mks.com/`
18. AlturaSoluciones: SIMPLe. (2010), `http://www.alturasoluciones.com/`
19. Kelemen, Z.D., Balla, D.K., Bóka, G.: Quality Organizer: A Support Tool In Using Multiple Quality Approaches. In: 8th International Carpathian Control Conference (ICCC 2007), Strbské Pleso, pp. 280–285 (2007)
20. Pardo, C., Pino, F.J., García, F., Piattini, M.: Tool for supporting the harmonization process - HProcessTool (2010), `http://alarcosj.esi.uclm.es/armonizacion/login.jsp`
21. Pardo, C., Pino, F.J., García, F., Piattini, M., Baldassarre, M.T.: Trends in Harmonization of Multiple Reference Models. In: Evaluation of Novel Approaches to Software Engineering. LNCS. Springer, Heidelberg (in press, 2011) (Special edition best papers ENASE 2010, extended and updated paper)
22. Gilbert, D.: JFreeChart project (2010), `http://www.jfree.org/jfreechart/`
23. Lemus, S.M., Pino, F.J., Piattini, M.: Towards a Model for Information Technology Governance applicable to the Banking Sector. In: V International Congress on IT Governance and Service Management (ITGSM 2010), Alcalá de Henares, pp. 1–6 (2010)
24. Pardo, C., Pino, F.J., García, F., Piattini, M., Baldassarre, M.T., Lemus, S.: Homogenization, Comparison and Integration: A Harmonizing Strategy for the Unification of Multiple-Models in the Banking Sector. In: The 12th International Conference on Product Focused Software Development and Process Improvement (PROFES 2011), Bari (in press, 2011)

Requirement Conflicts Resolution: Using Requirement Filtering and Analysis

Wasi Haider Butt, Sameera Amjad, and Farooque Azam

Department of Computer Engineering
College of Electrical & Mechanical Engineering
National University of Sciences & Technology, Pakistan
{wasi,sameera,farooq}@ceme.nust.edu.pk

Abstract. Varied frameworks of Requirement Engineering have been proposed by numerous research works majority of them focusing on the process as a whole. Few have managed to look beyond the basic framework, identifying complexities that lie within each activity of this crucial framework. One such complexity often faced by requirement engineers involves resolution of conflicts between the elicited set of requirements which are related to one another. This research provides with a conflict resolution strategy to overcome this complexity.

OBJECTIVE-The objective of this research is to present a systematic approach towards resolving software requirement spanning from requirement elicitation to requirement analysis activity of the requirement engineering process.

PROPOSED MODEL-Proposed in this study is a requirement "Conflict Resolution Strategy" (CRS) which employs requirement filter and an analysis strategy for resolving conflicts arising during software development.

METHODOLOGY- This model is based on both empirical results that we carried out and our extensive research on requirement conflict resolutions and the area of Requirement Engineering. Our model is based on filtering the elicited requirement set into three sets of requirements according to their nature of implementation and then analyzing them by applying CRS methodology.

CONCLUSION-Initial validation of the model indicates that the model is effective in identifying and resolving software conflicts that arise during a requirement engineering process.

1 Introduction

Requirement engineering necessitates classification of elicited requirements based on their nature of implementation and the analysis of relationships among these require-ments. Conflicts often arise while eliciting these requirements from different stake-holders who have varying interests and perspectives with respect to the product; detection and resolution of these conflicts is mandatory making analysis

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 383–397, 2011.

of requirements as the mainstay of requirement engineering process. If no subset of requirement(s) of a given stakeholder interfere with requirement(s) of another involved stakeholder then the Software Requirement Specification document is said to be stakeholder coherent SRS [1]. Inaccurate, ambiguous, and conflicting requirements are most common reasons for cost and schedule overruns of software development failing to satisfy user expectations in terms of functional and quality aspects of the delivered system [2]. Requirement Engineering is not just a single phase of software development process rather it is a complex process comprising of different sub activities that together lay the foundation of a software product on which software development of a product commences [3]. Requirement problems are widely responsible for the quality and effectiveness of the software [4] making their elicitation and analysis a critical pair of activities of the RE phase.

This paper identifies some core areas from RE research literature to develop a deeper understanding of conflicts that arise between software requirements during RE process. This research investigates the identification of conflicts, their classification based on the nature of their criticality and their treatment before exiting the RE process to prevent propagation of these conflicts; minimizing their effect in later phas-es of software development process. A conflict oriented requirement process has been proposed called μ**- Strategy** comprising of two major activities of the RE process for conflict resolution starting from requirement elicitation and spanning to the require-ment analysis activity. Both activities are further organized into a predefined set of sub activities that together ease conflict identification and resolution for a requirement engineer. Mode of validation used for studying effective of - Strategy involves a case study conducted during software development process of a module for a University Management System.

Section 2 provides a coarse grained overview of the RE process. Section 3 presents some research methodologies studied as part of literature. Section 4 explains the working of the μ- Strategy framework. Section 5 presents the case study results that evaluate behavior of the μ- Strategy to develop a customer consistent software requirement specification. Section 6 gives conclusion and future work for this research.

2 Background and Motivation

Requirements engineering covers all of the activities involved in discovering, documenting, and maintaining a set of requirements for a software system. The use of the term 'engineering' implies that systematic and repeatable techniques should be used to ensure that system requirements are complete, consistent and relevant. Zave [5] defines RE as a domain of software engineering that studies real world objectives and problems in achieving the desired outcome of a product; developing a suitable solution by studying precise specifications of the software under development. In other words precise and unambiguous requirements are key objectives in developing software whose behavior is consistence. This necessitates a conflict resolution strategy which controls and monitors divergence in the behavior of a software. Software Engineering literature classifies RE process into the following major activities:

- Requirement Elicitation: Developing requirement scope, requirement understanding and requirement volatility over discussion with customer.
- Requirement Analysis: This activity analyzes requirements against ambiguities, inconsistencies and conflicts. Requirement conflicts are widely responsible for risks associated with incorrect requirement implementation making their identification and resolution mandatory.
- Requirement Specification: This activity is responsible for transforming the above elicited and analyzed requirements into a requirement specification document.
- Requirement V&V (Verification & Validation): The requirement specification document is examined to verify conformance of requirements to the stakeholder needs. Next these requirements are tested for execution.

Requirements are statements of stakeholders regarding the system under development that when implemented satisfy user needs. Further, a single software requirement can consist of many parts. Problems arise when requirement(s) or part of a requirement(s) is inconsistent with or contradicts with other requirement(s). Such requirements are called conflicting requirements and can lead to fatal feature or quality failures of a software product. Absence of a formal conflict resolution process can lead to dissatisfaction of stakeholders eventually ending up in failure of the resultant system. Barry Boehm and Hoh In [7] presented numerous case studies which suffered drastic failures due to requirement conflicts of the developed systems. The purpose of this research is to investigate the existing methods and determine a suitable framework for conflict resolution that is easy to understand and implement by a requirement engineer having limited conflict resolution expertise.

Our studies of some existing methodologies for resolving requirement conflicts have brought forth some methods: the WINWIN System and CORA framework proposing different techniques for identifying and resolving requirement conflicts.

2.1 The WINWIN System

Barry Boehm and Hoh In used a knowledge based tool called the Quality Attribute Risk and Conflict Consultant (QARCC) that operated on the WinWin system working on the following principles:

- Identify and negotiate quality attribute requirement conflicts and tradeoffs
- Diagnose quality attribute conflicts on the basis of early information.

A knowledge base approach illustrated in Figure 1 was used by the QARCC tool to identify the conflict from past knowledge and then providing different options and suggestions to resolve these conflicts. Based on the knowledge collected using the WinWin negotiation system, the QARCC tool prioritized requirements into a hierarchy valued by the different classes of stakeholders [7]. This tool assisted the stakeholders to identify and negotiate risks associated with requirement conflicts and helped resolve them using negotiation.

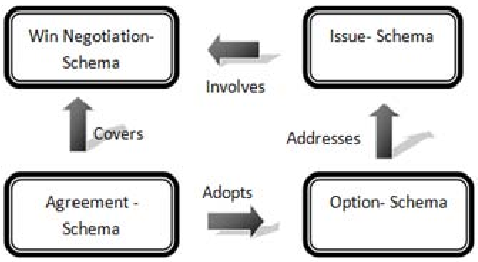


Fig. 1. A WinWin Negotiation QARCC Tool Technique

2.2 The Conflict-Oriented Requirement Analysis Framework

William N. Robinson [1] proposed a general framework for requirement conflict called Conflict Oriented Requirement Analysis (CORA) framework. Like many other studies, CORA targeted requirement conflict resolution using requirement ontology. CORA basic process is depicted in Figure 2 which follows a cycle of activities for requirements analysis [1]. CORA starts from requirement analysis phase of the RE process following the CORA cycle to resolve the conflicts.

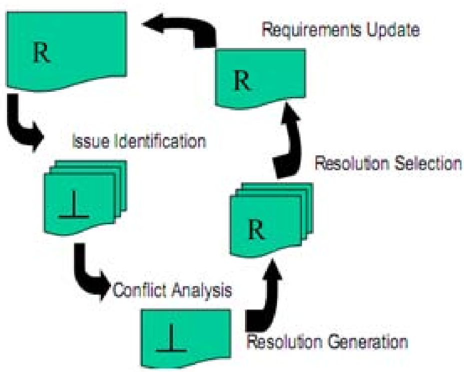


Fig. 2. CORA Process Activities [1]

3 Proposed Conflict Resolution Approach

We have used two major dimensions to resolve and manage conflicts using a strategy later referred to as the μ**- Strategy**. First dimension of this strategy includes classifying a hierarchy of requirements depending on the nature of their criticality which need to be resolved for conflicts. Unlike other existing models [8] which use requirement architectural decomposition, μ- Strategy focuses on both functional and non functional aspect of requirements for their classification. Second dimension includes operational implementation of the μ strategy that will be used to resolve conflicts from classified set of requirements. The μ requirement decomposition and its strategy are discussed below.

3.1 Nature of Requirements

Requirements can be broadly divided into two major types' namely functional requirements (FRs) and non functional requirements (NFRs). Every FR has one or more NFRs associated with it. Issues often arise due to technical or stakeholder conflicts preventing successful implementation of requirements [1]. These undesirable interactions or contradicting elements among requirements hinder mutual satisfaction of requirements producing conflicting set of requirements that lead to divergent system behavior. Overlapping or conflict parts of a requirement can be better illustrated in Figure 3 below.

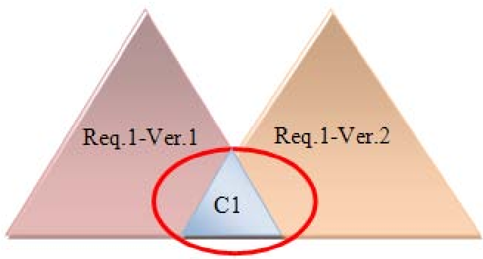


Fig. 3. Conflicting Requirements

Conflicts in functional requirements mainly arise due to participation of multiple stakeholders having varying system interests. Such conflicts are even more obvious to incur for non functional or quality requirements of a system. Conflicts in functional set of requirements can lead to immediate failure of the operational features of the system while conflicts in quality requirements of a system can lead to dissatisfaction on part of the user which disintegrates system performance. Hence, functional and non functional requirements both need to be resolved to prevent divergent system behavior.

3.2 Proposed μ-Strategy

The μ-Strategy (MEO-Strategy where MEO stands for mandatory, essential and optional breakdown of requirements) begins from requirement elicitation activity of the RE process. It structures elicited requirements into three basic types of requirements based on the nature of their implementation. μ-Strategy works on the below listed set of requirement observations:

1. Mandatory Requirements (MR): Primary set of functional and non functional requirements that are crucial for the success of a system and cannot be compromised.

2. Essential Requirements (ER): Secondary set of requirements include requirements that directly constraint the mandatory requirements providing support to the primary requirements of the system.
3. Optional Requirements (OR): These are requirements that can be implemented in such a way that the resultant conflict has no effect on the acceptance.

The μ-Strategy does not venture into the details of the different types of functional and non functional requirements as ample literature is already available in that regard focusing majorly on the relationship of requirements both FRs and NFRs that are necessary for success of the software. Since both are equally important for the successful deployment of a software system, therefore, it just classifies them broadly into the above MR, ER, and OR classification. μ-Strategy applies its conflict resolution strategy on requirements starting from the requirement elicitation phase which is where these conflicts actually get injected into the RE process. μ-Strategy follows following sub activities to accomplish conflict resolution:

1. Requirement Filtering: Filter requirements as requirements get elicited depending on the nature of their criticality.
2. Conflict Prevention: Prevent conflicts in mandatory requirements while instantiating requirements. Document these requirements resolving conflicts as and when they arise in the presence of stake holders.
3. Conflict Detection & Removal: Requirements that enter requirement analysis sub activity are analyzed and detected for conflicts. These conflicts are negotiated [11] and resolved using agreed upon alternatives.
4. Conflict Containment: Effect of conflicts identified in these requirements can be neutralized during implementation as their containment has little or no impact on User Acceptance Test.

The μ-Strategy adopts below depicted phenomena (illustrated in Figure 3) for resolving conflicts starting from elicitation phase and spanning to requirement analysis which marks end of conflict resolution. Requirement conflicts in mandatory and essential requirements are resolved by the end of Elicitation and Analysis activities of the RE process while optional requirements are left for later resolution owing to delay in their implementation. That is,

1. Conflicts are prevented in the mandatory requirements (resolved while requirement elicitation itself)
2. Conflicts are detected and analyzed in requirement analysis phase and are resolved using the divide & Conquer principle of NFD [8].
3. Conflicts in optional requirements are contained.

μ-Strategy Overview. μ-Strategy as illustrated in Figure 5 below begins from requirement elicitation phase of the RE process and follows a cycle of the below listed steps to resolve conflicts that may arise in requirements starting from elicitation and spanning to the requirement analysis activity. After filtering

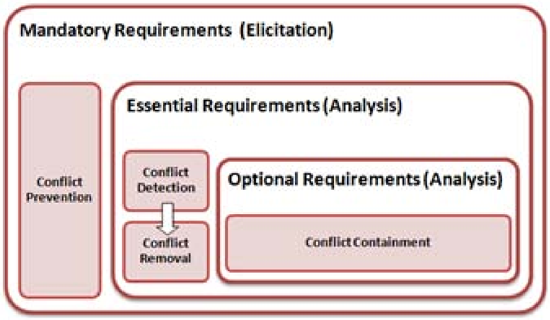


Fig. 4. Basic concept of μ-Strategy

requirements as they are elicited, conflicts are pre-vented in mandatory requirements as they are most crucial for fulfillment of primary features of the software system. Essential requirements and optional requirements are passed on to the requirement analysis where these requirements are further analyzed. Essential requirements are analyzed using divide and conquer principle and then are resolved for identified conflicts through negotiations with the participating stakeholders. Optional requirements are not resolved for their conflicts neutralizing their affect they are documented in the Software Requirement Specification (SRS) as they do not impact system divergent behavior greatly and can be executed while implementation canceling the effect of one another. MR, ER and OR are documented in the SRS after which they are passed on to the validation phase of the RE process for testing their execution. This process has been illustrated in the below given -Strategy workflow when working within a RE process.

The μ-Elicitation Strategy. Requirement elicitation plays a critical role in the foundations of a requirement specification document and it is here where the conflicts actually get infused into the requirement document while drawing requirements from different stakeholders. We filter requirements as and when they are elicited (namely into MR, ER and OS requirements). After filtering requirements, we prevent conflicts in requirements that are directly responsible for contributing into the system goals or yield direct value to its users leaving minor conflicts that have minor impact on actual implementation.

Identification of such primary FR and NFR is similar to identifying processes deemed most important for the success of an organization. An example of mandatory functional requirement conflict includes; stakeholder A states a mandatory requirement such as" the exam cell coordinator must be able to generate a cumulative total of individual PhD students and assign them grades accordingly". Another stakeholder B states a requirement such as "the exam cell coordinator must be able to generate a cumulative result of PhD students who have cleared the GRE test" that contradicts the previous version of the same requirement. Since both stakeholders are participants of the RE process, therefore, the RE engineer will negotiate with both stakeholders to prevent the contradicting

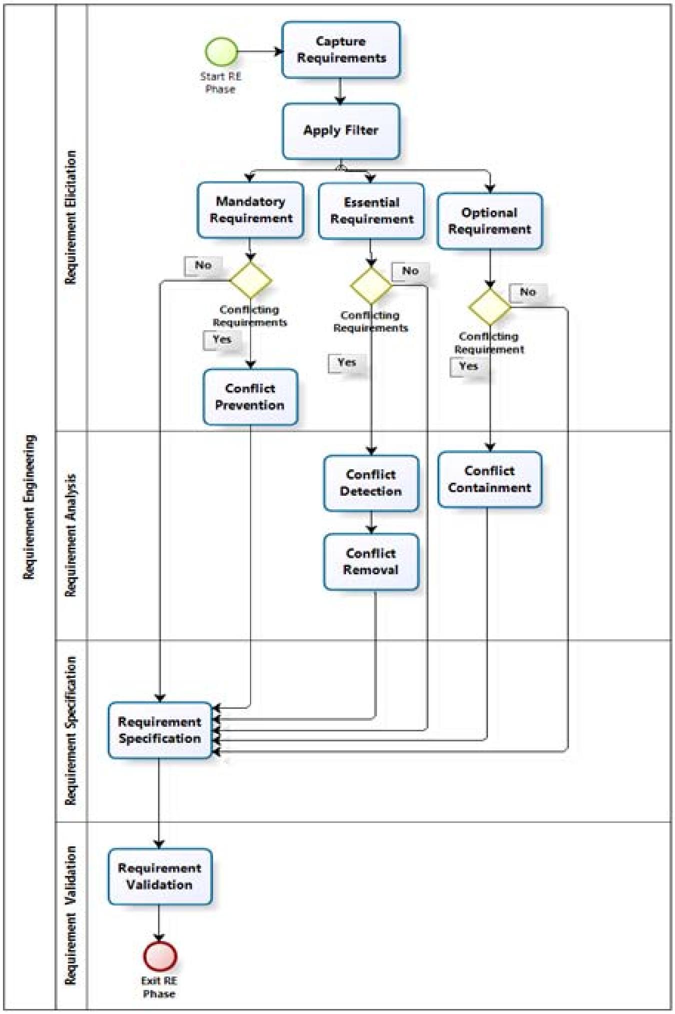


Fig. 5. Basic μ-Strategy for Conflict Resolution in a RE Process

statements that are form major feature of the Exam Management System of a university. One perspective of the two stakeholders is selected for implementation after agreement of both stakeholders through negotiations as illustrated in Figure 6. For example, the perspective that maps with the policy of the University for grading eligibility will be used after negotiation. We let the essential and optional set of requirements to get elicited in the requirement elicitation activity and enter into the requirement analysis phase for conflict analysis rather than resolving them here.

Fig. 6. Conflict Prevention for Mandatory Requirements

The -Analysis Strategy. Essential requirements define functionality (FR and NFR) that are secondary to the goals of the system mainly consisting of non functional requirements. These requirements are those without which a system cannot pass acceptance testing in its operational environment. Essential requirements are similar to secondary requirements [8]. Examples include functions that aid the mandatory requirements like managing them or the data generated by the system making conflict resolution of such requirement necessary. Different conflict resolutions have been proposed in research literature [12]. Conflict resolution strategy used in our proposed framework is illustrated in Figure 7 for above identified requirements is as follows:

1. Requirements are analyzed for conflicts.
2. Conflicting parts of ER are divided using the Divide-and-Conquer principle [8] which states that if a requirement has parts that contradict or conflict with parts of other requirements then it is useful to separate these parts and resolve the conflicts.
3. Using Divide and Conquer, conflicts are detected and resolved through negotiations using suggested alternatives. Since majorly these requirements are quality oriented NFRs therefore these can be resolved for conflicts by dividing them and then integrating both requirements by combining them and achieving a common perspective for its implementation.

For example, "after three incorrect login attempts the system should block the user account which being tried to gain access", related conflict "system should never block the account of a user for incorrect log in attempts". Since, ER is necessary set of requirements providing support to the mandatory requirements their fulfillment is necessary. Hence, by applying the above strategy of Conflict Detection and Removal; by the end of requirement analysis conflicts in ER are completely resolved. Optional requirements (OR) are requirements which can be contained and do not have an impact on the deployment of the system or on user acceptance tests conducted in operational environment. For example, if Stakeholder A has a requirement that means changing the entire layout of the system user interfaces. This would mean following the -Strategy cycle again, negotiating with other involved stakeholders who may have different opinions and

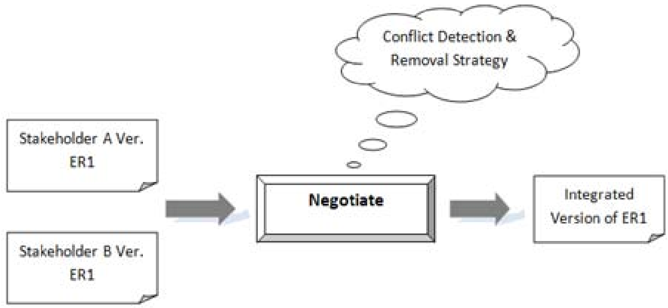


Fig. 7. Conflict Detection & Removal for Essential Requirement

making amendments. When the cost of resolving such a conflict of additional requirements is greater than the benefit of the requirement implementation then we group such requirements under OR group and contain their effect while implementation or delay it for future considerations in upcoming versions of the product under development. Conflict containment is depicted in Figure 8 below.

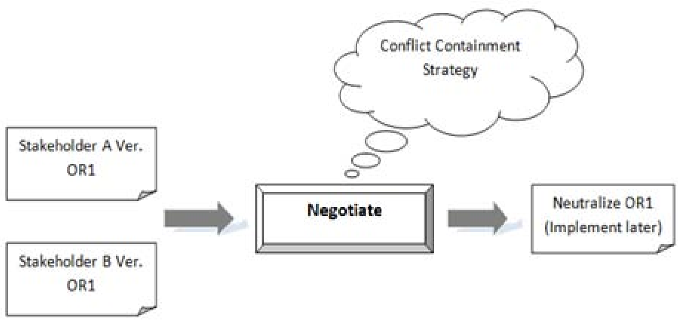


Fig. 8. Conflict Containment for Optional Requirement

4 Model Validation

In order to check and validate the practical functioning and impact of the proposed approach, the μ-Strategy was applied during the requirement phase on one of the modules of university management system (UMS) which was under development in our own university. Following are the details of implementation, results and the findings. Plan was to automate all the business processes running in the university like academics, administration, accounts and so on. For the validation of our conflict resolution strategy, we selected Hostel Management System which was a subpart of the large university management system. A team of six people consisting of a team lead, two developers, a tester and two requirement engineers started development within the campus development cell. Time assigned for the completion of version 1.0 for the module was one month.

For requirement elicitation, open interviews were held with stakeholders as this elicitation mode most suited the development environment. The stakeholders identified included the hostel provost, wardens, supervisors, top management and the faculty. In the previous manual process, Provost used to manage all hostels, their capacity, adding deleting space to hostels etc. Hostel warden s were supposed to assign students hostels or vacate hostel seats. In this article, 24 requirement samples have been extracted from the initial requirement draft.

Warden A, was interviewed the first day and the following requirements were elicited:

RA 1. System should allow warden to assign student a seat in his hostel
RA 2. System should allow warden to shuffle multiple students seats
RA 3. System should maintain a log of all allotments and vacations
RA 4. System should allow warden to see log of allotments and vacations of his hostel
RA 5. System should allow warden to check trend of which class students like to live with which class students.
RA 6. System should allow warden to put a student in waiting list if a seat is not available and student requests for a seat
RA 7. System should not allow warden to assign a student a seat from outside the waiting list if there is any student in the waiting list
RA 8. System should allow warden to check the waiting list
RA 9. System should not allow warden of any other hostel or provost to interfere his hostel matters
RA 10. System should allow students to register their complaints
RA 11. System should allow warden to check those complaints
RA 12. System should allow warden to check those complaints in date wise order
RA 13. If any user other than the designated warden tries to access a specific hostel information, system should block his account immediately.
RA 14. Warden B, was interviewed the second day and the following requirements were elicited:
RA 15. System should allow warden to assign student a seat in his hostel if he has approval from provost.
RA 16. System should maintain a log of all allotments and vacations
RA 17. System should allow warden to see log of allotments and vacations of his hostel
RA 18. System should allow warden to put a student in waiting list if a seat is not available and student requests for seat with provosts approval
RA 19. System should allow warden to assign a student a seat from the waiting list or from outside the waiting list if he has approval from management
RA 20. System should allow warden to check the waiting list
RA 21. System should allow management, provost to view status of all hostels
RA 22. System should allow student to register their complaints

RA 23. The registered complaints should be visible to all wardens, provost and the management.
RA 24. The complaints should be viewed in any selected order that user should be allowed to select from a list etc.
RA 25. Top management, provost should be allowed to view records of each hostel to check all wardens performance.

After detailed negotiations with the wardens, provost and management, following ranking of requirements and conflicts were managed by applying our proposed approach.

Table 1. Requirements Matrix

Requirement Id	Type	Has Conflict	Conflict With	Solution
RA 1	MR	Yes	RB 1	CP
RA 2	OR	No		
RA 3	ER	No		
RA 4	ER	No		
RA 5	OR	No		
RA 6	ER	Yes	RB 4	CD
RA 7	ER	Yes	RB 5	CD
RA 8	ER	No		
RA 9	ER	Yes	RB 7	CD
RA 10	MR	No		
RA 11	ER	Yes	RB 9	CD
RA 12	OR	Yes	RB 10	CC
RA 13	ER	Yes	RB 11	CD
RB 1	MR	Yes	RA 1	CP
RB 2	ER	No		
RB 3	ER	No		
RB 4	ER	Yes	RA 6	CD
RB 5	ER	Yes	RA 7	CD
RB 6	ER	No		
RB 7	ER	Yes	RA 9	CD
RB 8	MR	No		
RB 9	ER	No		
RB 10	OR	Yes	RA 12	CC
RB 11	ER	Yes	RA 12	CD

Mandatory Requirements (MR): Essential Requirements (ER): Optional Requirements (OR): Conflict Prevention (CP): Conflict Detection & Removal (CD): Conflict Containment (CC):

After treating the above conflicts, requirement draft was finalized. The conflict that arose in mandatory was prevented from entering in to the final requirement draft for instance the conflict that arose between RA 1 and RB 1 was prevented by developing a consensus among provost, management and all the wardens. Before writing down RB 1 that was extracted after elicitation of RA 1, it was checked against previously extracted requirements since it was a mandatory requirement. If conflict was found, before writing down RB 1, the conflict was resolved; this is where conflict was pre-vented in the first place from getting infused into the system development cycle. Similarly the requirements that were essential were jotted down as they were elicited without checking them against the written set of requirements but when the require-ments were analyzed during analysis, conflicts were detected and removed by nego-tiating with the stakeholders. Lastly, the optional requirements were left untreated and their effect was neutralized during their execution. We can see clearly that the conflict between RA 12 and RB 10 has no impact on implementation and acceptance since different options for sorting the list are provided; date wise sorting can be chosen as default sorting to prevent disagreement of stakeholders who can choose their own desired behavior of the system.

After requirements process, system was developed, tested and deployed successfully. Followed by this successful operation of the deployed system, a feedback workshop was conducted in which feedback about system was collected from users. Following graph depict the level of user satisfaction. From the case study graphs below it is clear that the user acceptance test for system performance, quality and conformance to user needs have been achieved successfully, with all system impacting conflicts being identified and treated by μ-Strategy.

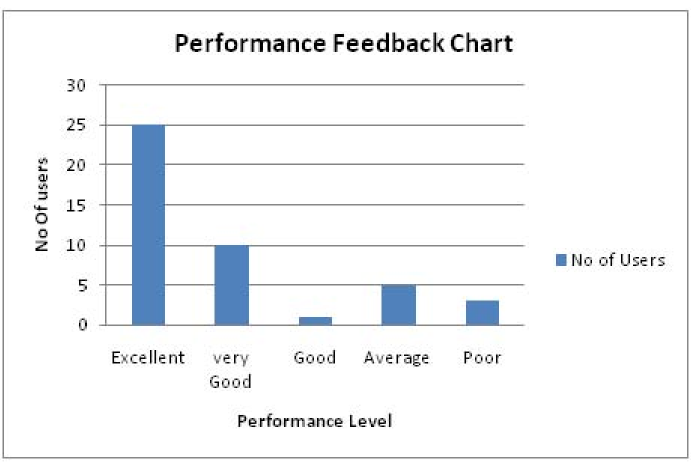


Fig. 9. User Acceptance Performance Feedback Chart

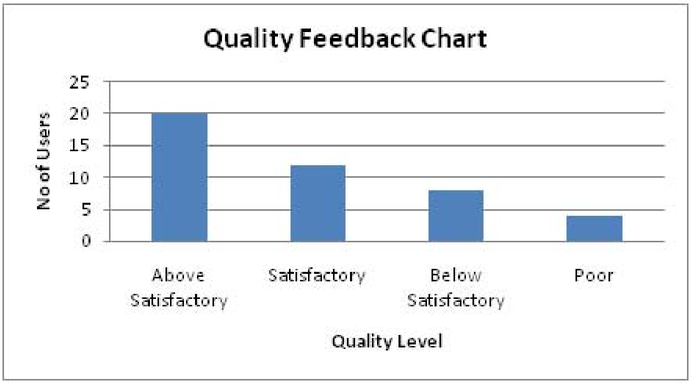


Fig. 10. User Acceptance Quality Feedback Chart

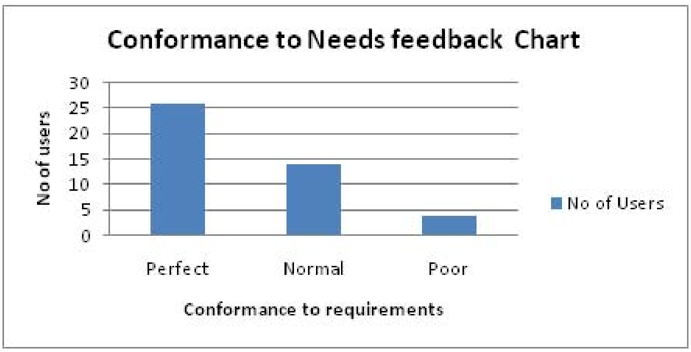


Fig. 11. Requirement Conformance Feedback Chart

5 Conclusion and Future Work

Conflicts pose a great challenge to the requirement engineering process of a software development framework. These conflicts hinder successful deployment of the software leading to user acceptance test failures. The objective of this research was to propose a conflict resolution framework that identifies different types of conflicts and suggests resolution techniques to resolve each type of conflict. Three groups of requirements were identified that needed to be targeted to overcome conflicts in a requirement specification. Research literature was studied to identify these requirements that were more liable to conflicts and a Conflict resolution strategy was proposed to resolve these conflicts. A case study was conducted in which the proposed model was employed while the RE process to identify and resolve different types of conflicts that arise in a real world environment. The results of this case study indicated successful identification and resolution of different types of conflicting requirements involving stakeholders from different perspectives and interests.

The concept of using a simulation in this regard is appealing to our future work. It involves identifying other existing conflict resolution techniques (not discussed in this paper) and simulating a virtual environment validating and comparing performances of these varying models to identify the position of -Strategy against these models. Since practically it is time consuming and risky to employ existing conflict resolution models for an RE process, it was thought wise to use a simulation for achieving our future goal to study this challenging area.

References

1. Robinson, W.N., Volkov, V.: Requirement Conflict Restructuring. Conflict-Oriented Requirement Restructuring, GSU CIS Working Paper 99-5, pp. 1–47 (1999)
2. Khalid, E.E., Nazim, M.H.: A Field of Requirements Engineering Practices in Information Systems Development. In: Second International Symposium on Requirements Engineering, pp. 68–80 (1995)
3. Sommerville, I., Sawyer, P., Viller, S.: "Improving the Requirements Process. In: Fourth International Workshop on Requirements Engineering: Foundation of Software Quality, pp. 71–84 (1997)
4. Sommerville, I.: Software Engineering, 5th edn. Addison-Wesley, Reading (1996)
5. Zave, P.: Classification of research Efforts in Requirement Engineering. ACM Computing Surveys 29(4), 315–321 (1997)
6. Pressman, R.S.: Software Engineering: A practitioner's Approach, 5th edn. McGraw-Hills Publication, New York (2001)
7. Boehm, B., In, H.: Identifying Quality- Requirement Conflicts. In: Proceedings of IEEE Software, pp. 26–35 (March 1996)
8. Poort, E.R.: Proceedings of the Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA 2004), pp. 1–10 (2004)
9. Dube, R.R., Dixit, S.K.: Process-oriented Complete Requirement Engineering Cycle for Generic Projects. In: International Conference and Workshop on Emerging Trends in Technology (ICWET 2010) -TCET, Mumbai, India, pp. 194–197 (2010)
10. Gilb, T.: Principles of software Engineering Management. Addison-Wesley, Reading (1988)
11. Jelassi, M.T., Foroughi, A.: Negotiaition Support Systems: An Overview of Design Issues and Existing Software. Decision Support Systems, 167–181 (1989)
12. Klein, M.: Supporting Conflict Resolution in Cooperative Design Systems. IEEE, Transactions on Systems, Man and Cybernetics, 1379–1390 (1991)

Efficient Algorithm for Detecting Parameterized Multiple Clones in a Large Software System

Rajesh Prasad[1], Suneeta Agarwal[1], Anuj Kumar Sharma[1], Alok Singh[1], and Sanjay Misra[2]

[1] Motilal Nehru National Institute of Technology
Allahabad-211004, India
[2] Department of Computer Engineering
Federal University of Technology
Minna, Nigeria

Abstract. Two code fragments are said to be similar if they are similar in their program text or in their functionalities. The first kind of similarity can be detected with the help of *parameterized string matching*. In this type of matching, a given pattern P is said to match with a sub-string t of the text T, if there exists a bijection between the symbols of P and the symbols of t. The parameterized string matching problem has been efficiently solved by Fredriksson and Mozgovoy by using the shift-or (PSO) algorithm. The drawback of this algorithm is: it is unable to handle patterns of length greater than the word length (w) of a computer. In this paper, we solve this word length problem in a bit-parallel parameterized matching by extending the BLIM algorithm of exact string matching. Extended algorithm is also suitable for searching *multiple patterns* simultaneously. Experimentally, it has been observed that our algorithm is comparable with PSO for pattern length ≤ w and has ability to handle longer patterns efficiently.

Keywords: Parameterized string matching, bit-parallelism, BLIM, software maintenance, clone detection and multiple patterns.

1 Introduction

Traditional string matching problem is to find all the occurrences of a pattern P[0…m-1] in the text T[0…n-1], where symbols of P and T are drawn from some finite alphabet Σ of size σ. The detail about string matching algorithm can be found in [5, 7, 9, 11, 12, 13]. Application area of string matching includes: computational biology, information retrieval, plagiarism detection, similarity detection etc. There are basically two kinds of similarities between two code fragments (i) Similarity based on their program text and (ii) Similarity based on their functionalities. The first kind of similarity is often the result of copying a code fragment and then pasting to another location. In [1], classification of clones based on textual and functional similarities has been discussed. The clones based on the textual similarities [1] are classified as: Type I, Type II and Type III clones and clones based on the functional similarities are Type IV clone. Type II clones are obtained by renaming identifiers, literals and

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 398–409, 2011.

variables. *Parameterized string matching* [3, 8] has been applicable in detecting Type II clones, which are helpful in *software maintenance* [3] *and plagiarism detection* [6]. In the parameterized string matching problem [3, 4], two disjoint alphabets are used: (i) Σ: for fixed symbols alphabet (ii) Π: for parameter symbols alphabet. The symbols of *pattern* P[0…m-1] and *text* T[0…n-1] are taken from Σ∪Π. In this matching, while looking for occurrences of P in T, the symbols of Σ must match exactly whereas the symbol of Π can be consistently renamed. A given pattern P is said to match with a sub-string t of the text T, if ∃ a one-to-one correspondence between symbols of P and symbols of t. For example, by defining the mapping A↔T and C↔G, the sequences ACATG and TGTAC are textually similar. Only few algorithms [3, 4, 6, 8, 10] exists for parameterized matching. In [3], parameterized on-line matching algorithm for a single pattern was developed. This algorithm runs in $O(n \log \min(m, |\Pi|))$ worst-case time. In [6], a bit-parallel string matching algorithm (PSO) and BDM string-matching algorithm for parameterized string matching problem was developed. In [8], backward non-deterministic DAWG (BNDM) for parameterized matching was developed. This algorithm was shown to be faster than PSO [6]. In [10], a fast parameterized string matching with q-gram was developed. This algorithm was shown to be very efficient for specific value of q. In [3], two alphabets: fixed and parameterized alphabets were used. In [10], only one alphabet, called parameterized alphabet was used (here Σ = ϕ}. The algorithm (FPBMH) in [10] was shown to be sub-linear on average and has been extended for two dimensional parameterized string matching.

The main problem in bit-parallel algorithms [2, 6, 8] is that they are unable to handle patterns of length greater than word length (w) of a computer, efficiently. This problem can be solved by either partitioning the pattern into number of parts, each of size ′w′, or considering array of multiple words. Either of these techniques has not been useful in recent years as this degrades the performance of any bit-parallel algorithms. In [14], an efficient algorithm (BLIM) for solving word length problem in exact string matching has been developed. This algorithm was shown to be comparable with other bit-parallel algorithm when $m \leq w$ and was shown to have good performance when $m > w$.

In this paper, we extend the BLIM (Bit-parallel Length Independent Matcher) algorithm [14] of exact string matching for *parameterized string matching*, to solve the word length problem in a bit-parallel parameterized string matching algorithms. We call the extended algorithm as PBLIM. We further extend the PBLIM algorithm for handling *multiple patterns* [2, 6, 14, 15]. We compare the performance of PBLIM with existing PSO algorithm [6] for two different cases: one for $m \leq w$ and other for $m > w$. For $m > w$, we implement PSO algorithm [6] by considering the array of multiple words. Experimentally, it has been observed that, for $m \leq w$, PBLIM is comparable with PSO [6] and for $m > w$ PBLIM has very good running time as compared to PSO [6]. For experiment purpose, we have taken a DNA file as a text and any substring of that DNA as a pattern.

The paper is organized as follows. In Sec. 2, we present the related algorithms for exact string matching problem. In Sec. 3, we present the related algorithms for parameterized string matching problem. In Sec. 4, we present our proposed algorithm (PBLIM) for parameterized string matching that can overcome word size limitation in a bit-parallel parameterized string matching. In Sec. 5, we present experimental results. Finally, we conclude in Sec. 6.

2 Algorithms for Exact String Matching

2.1 Shift-or Algorithm

In this section, we present standard shift-and [2] string matching algorithm for exact string matching problem. First we define the following terms: (i) $b_{w-1}b_{w-2}\ldots b_1b_0$ denotes bits of computer word of length w. (ii) Exponentiation is used to denote bit repetition (e. g. 0^41=00001). C-like syntax is used for operations on the bits of computer words: "|" is for bit-wise or, "&" is for bit-wise and, ^ is bit-wise xor, "~" complements of all the bits. The shift left operation, "<<r", moves all bits to the left by ′r′ and enters ′r′ zeros in the right. Σ is an alphabet of size σ.

In the shift-and algorithm [2], a non-deterministic finite automata (NFA) automaton of a given pattern P[0..m-1] is constructed in the preprocessing phase. The automaton has states 0, 1, 2…m, with state 0 as initial state, state m as final state and $\forall$ i = 0...m-1, there is a transition from state i to state i+1 for character P[i] of the pattern P. In addition, there is a transition for every c $\in$ Σ from and to the initial state. This algorithm builds a table B having one bit mask entry for each c $\in$ Σ. For $0 \leq i \leq m-1$, the mask B[c] has i^{th} bit set to 1 iff P[i] = c otherwise it is 0. If the i^{th} bit in B[c] is 1, then in the automaton, there is a transition from the state i to i+1 with character c. For searching, algorithm needs a bit mask D so that i^{th} bit of this mask is set to 1, if and only if state i in NFA is active. For each text symbol c the state vector D is updated by $D \leftarrow ((D << 1) \mid 1)$ & B[c], where D << 1, makes the state active and then looks for transition on that state. If after the i^{th} step (after processing the i^{th} symbol of the text), the $(m-1)^{th}$ bit of D is 1, then there is an occurrence of P with shift i-m. If $m \leq w$, then the running time of the algorithm is $O(n)$.

2.2 The BLIM Algorithm

This section presents the existing BLIM (bit-parallel length independent matching) [14] algorithm for exact string matching that does not restrict the pattern length (m) to be less than the word size (w). Given a pattern P, let's imagine an alignment matrix A, where each row_i, $0 \leq i < w$, contains the pattern right shifted by i characters. Thus A has w rows and ws = w + m - 1 columns. In the present discussion, 'ws' value will be referred as window size. A sample alignment matrix corresponding to the pattern P = abaab over Σ = {a, b, c, d} is shown in Table 1, assuming computer word size w = 8. The main idea of BLIM is to slide that 'ws' length alignment matrix over the text, check if any possible placements of the input pattern exists in the current window via bitwise operations, and after completing the investigation of the current text portion T[i…i+ws - 1], shift the window to the right by an amount that is computed according to the immediate text character T[i+ws] following the current window.

Table 1. Alignment Matrix

	0	1	2	3	4	5	6	7	8	9	10	11
0	a	b	a	a	b							
1		a	b	a	a	b						
2			a	b	a	a	b					
3				a	b	a	a	b				
4					a	b	a	a	b			
5						a	b	a	a	b		
6							a	b	a	a	b	
7								a	b	a	a	b

When the window is located on text T[i…i + ws-1], BLIM visits the characters T[i + j] ($0 \le j < ws$) in an order that was previously computed at the preprocessing stage. At each character visit, possible occurrences of the pattern are checked with a bitwise AND operation by using a mask matrix that was again pre-computed. Current window is slid right by the shift amount specified by the text character T[i+ws] after the current investigation is over. Thus, a shift vector of alphabet size needs to be calculated at preprocessing also. In summary, in preprocessing phase, BLIM calculates the mask matrix, the shift vector and decides on the scan order according to input pattern. The detail can be found in [14].

To deal with *multiple patterns* [14], the alignment matrix (Table 1) incorporates all the patterns being considered, simultaneously. Let the patterns set be P_0, P_1, P_2……P_{r-1}, where r is the number of patterns. Length of each pattern is denoted by m_i, where $0 \le i \le r$. The length of longest and smallest patterns is denoted by m_{max} and m_{min} repectively. Window size in multi-patterns cases is denoted by $ws=Z+m_{max}-1$, where $Z = \lfloor w/r \rfloor$. Actually Z interprets the number of right shifted alignments for each pattern in the set. Assuming that the investigation window is aligned at text portion T[i...i+ws-1], the algorithm will check each pattern beginning at positions i to i+Z-1. The detail can be found in [14].

3 Algorithms for Parameterized String Matching

3.1 Parameterized String Matching Problem

This section presents small introduction to parameterized matching [3, 8] problem. Here we assume that all the symbols of P[0…m-1] and T[0…n-1] are taken from $\Sigma \cup \Pi$, where Σ is fixed symbol alphabet of size σ and Π is parameter symbol alphabet of size π. A pattern P matches the text substring T[j…j+m-1], for $0 \le j \le n-m$, if and only if $\forall\ i \in \{0, 1, 2,…m-1\}$, f_j (P[i]=T[j+i]), where $f_j(.)$ is a bijective mapping on $\Sigma \cup \Pi$. It must be identity on Σ but need not be identity on Π. For example, let P = XYABX on $\Sigma = \{A, B\}$ and $\Pi = \{X, Y, Z, W\}$. Pattern P matches the text substring

ZWABZ with bijective mapping X → Z and Y → W. This mapping can be simplified by *prev*-encoding [3]. For any string S, *prev*(S) maps its each parameter symbols s to a non-negative integer p, where p is the number of symbols since the last occurrences of s in S. The first occurrence of any parameter symbol in *prev*- encoding is encoded as 0 and if $s \in \Sigma$ it is mapped to itself (i.e. to s). For example, *prev*(P) = 00AB4 and *prev*(ZWABZ) = 00AB4. With this scheme of *prev*-encoding, the problem of the parameterized string matching can be transformed to the exact string matching problem, where *prev*(P) is matched against *prev*(T[j…j+m-1]), for $0 \leq j \leq n\text{-}m$. The *prev(*P) and the *prev*(T[j…j+m-1]) can be recursively updated as j increases with the help of following Lemma 1 [3].

Lemma 1: Let S′ = prev(S). Then for S″ = prev(S [j…j+m-1]) for all i such that S [i] $\in \Pi$ it holds that S″ [i] = S′ [i] iff S″ [i] < m, otherwise S″ [i] = 0.

3.2 Parameterized Shift-or (PSO)

In this section, we present parameterized shift-or (PSO) string matching algorithm [6]. The algorithm is generalization of the algorithm explained in section 2. It has been generalized in the following way:

(i) The pattern P is encoded by *prev*-encoding and stored as *prev*(P). In order to compute *prev*(P), an array *prev*[c] is formed, which for each symbol $c \in \Sigma$, stores the position of its last occurrence in P. For example, for the pattern P = XAYBX with Σ = {A, B} and Π = {X, Y}, *prev* (P) = 0A0B4.
(ii) For all j = 0, 1, 2...n-m, *prev* (T[j…j+m-1]) can be efficiently *prev*-encoded by Lemma 1.
(iii) The table B is built such that all the parameterized pattern prefixes can be searched in parallel. To simplify indexing into array B, it is assumed that Σ = {0, 1…σ-1}, and *prev*-encoded parameter offsets are mapped into the range {σ…σ+m-1}. For this purpose, an array *A*[0…σ+m-1] is formed, in which the positions 0…σ-1 are occupied by element of Σ and the rest positions are occupied by *prev*-encoded offsets. For example, for the pattern P = XAYBX on Σ = {A, B} and ∏ = {X, Y}, with σ = 2, m = 5, the *prev*-encoded pattern P is *prev*(P) = 0A0B4. The array *A* looks like as:

A	B	0	1	2	3	4
0	1	2	3	4	5	6

Searching for P′= *prev*(P) in T′ = *prev*(T) can’t be done directly as explained below. Let the pattern P = XAXX and the text T = ZZAZZAZZ. P′ = 0A21 and T′ = 01A21A21. Obviously, P has two overlapping parameterized occurrences in T (one with shift = 1 and another with shift = 4) but P′ does not have any occurrences in T′. The problem occurs because of when searching for all the m prefixes of the text in parallel, then some non-zero encoded offset p in T′ should be interpreted as zero in some case. For example, when searching for P′ in T′[1…4] = 1A21, 1(from left) should be zero. The solution to this problem is that, the lemma 1 should be applied in parallel to all m-length sub-strings of T′. This can be achieved in the following way.

The bit vector B[A[σ+i]] is the match vector for A[i], where $0 \le i \le \sigma+m-1$. If the j^{th} bit of this vector is zero, it means that P′[j] = A[i]. If any of the i^{th} least significant bit of B[A[σ]] is zero then corresponding bit of B[A[σ+i]] is also cleared. This can be achieved as: B[A[σ+i]] ← B[A[σ+i]] & (B[A[σ]] | (~ 0 << i)) which signifies that for $\sigma \le i \le \sigma+m-1$, A[i] is treated as A[i] for prefixes whose length is greater than A[i] and as zero for shorter prefixes thus satisfies Lemma 1.

4 Parameterized BLIM Algorithm (PBLIM)

In this section, we present our proposed algorithm (PBLIM) for parameterized string matching problem. This algorithm overcomes the word size limitation in a bit-parallel parameterized string matching algorithms. This algorithm is an extension of BLIM algorithm [14] discussed in section 3.

4.1 Preprocessing

The algorithm firstly calculates the *prev-encoding* [3] of the pattern to be matched and subsequently keep on calculating the prev-encode of each *ws* length window. In the preprocessing stage, we use two mask matrices: MASK and PMASK. The MASK matrix containing the entries for alphabets is computed as in BLIM algorithm [14]. In this matrix we need not compute the mask values for characters that are in the parameterized set: Π. The second matrix PMASK contains mask values of numbers from 0 to ws-1. The inherent problem that has to be overcome in parameterized string matching is that each numeric value 1 to ws-1 should also match with 0 in order to apply Lemma 1 of Baker [3] for calculating *prev* encoding of subsequent text. For example: Let us consider the pattern P = xax and text T = xaxax over Σ = {a} and Π = {x}. *prev*-encode of P, *prev*(P) = 0a2, *prev* encode of T, *prev*(T)= 0a2a2.

In this case there are two matches at offset 0 and 2 of text T, but the pattern being *prev*(P) will not match at offset 2 where the subtext is 2a2. So to overcome this problem, 2 should match with 0. To achieve this we logically "or" the mask value of number 0 with the corresponding column entries of every other numbers from 1 to ws-1. But this approach can generate potential false matches. For example: Let the pattern P = xax and text T = xay over Σ = {a} and Π = {x, y}. *prev* encode of P = 0a2 and *prev* encode of T = 0a0.

The above approach will wrongly match the pattern at offset 0 in text T. In PBLIM algorithm, we overcome this problem by selectively "OR"ing the bits of mask matrix (PMASK) according to its position in the window. Example 1 illustrates the process.

Example 1

Let the pattern P = abcba on Σ = {c} and Π = {a, b}, W = 8, m = 5, ws = 12, *prev* of P = 00c24. The mask matrix (MASK) for characters from Σ is shown in Table 2. Intermediate matrix (PMASK′) for calculating PMASK matrix is shown in Table 3.

Table 2. MASK matrix

	0	1	2	3	4	5	6	7	8	9	10	11
a	-	-	-	-	-	-	-	-	-	-	-	-
b	-	-	-	-	-	-	-	-	-	-	-	-
c	FE	FC	F9	F2	E4	C9	93	27	4F	9F	3F	7F
d	FE	FC	F8	F0	E0	C1	83	07	0F	1F	3F	7F

Table 3. Intermediate Matrix (PMASK′)

	0	1	2	3	4	5	6	7	8	9	10	11
0	FF	FF	FE	FC	F8	F1	E3	C7	8F	1F	3F	7F
1	FE	FC	F8	F0	E0	C1	83	07	0F	1F	3F	7F
2	FE	FC	F8	F1	E2	C5	8B	17	0F	1F	3F	7F
3	FE	FC	F8	F0	E0	C1	83	07	0F	1F	3F	7F
4	FE	FC	F8	F0	E1	C3	87	0F	1F	3F	7F	FF
5	FE	FC	F8	F0	E0	C1	83	07	0F	1F	3F	7F
6	FE	FC	F8	F0	E0	C1	83	07	0F	1F	3F	7F
7	FE	FC	F8	F0	E0	C1	83	07	0F	1F	3F	7F
8	FE	FC	F8	F0	E0	C1	83	07	0F	1F	3F	7F
9	FE	FC	F8	F0	E0	C1	83	07	0F	1F	3F	7F
10	FE	FC	F8	F0	E0	C1	83	07	0F	1F	3F	7F
11	FE	FC	F8	F0	E0	C1	83	07	0F	1F	3F	7F

Table 4. PMASK Matrix

	0	1	2	3	4	5	6	7	8	9	10	11
0	FF	FF	FE	FC	F8	F1	E3	C7	8F	1F	3F	7F
1	FF	FE	FC	F8	F0	E1	C3	87	0F	1F	3F	7F
2	FF	FF	FE	FD	FA	F5	EB	D7	AF	5F	BF	7F
3	FF	FF	FE	FC	F8	F1	E3	C7	8F	1F	3F	7F
4	FF	FF	FE	FC	F9	F3	E7	CF	9F	3F	7F	FF
5	FF	FF	FE	FC	F8	F1	E3	C7	8F	1F	3F	7F
6	FF	FF	FE	FC	F8	F1	E3	C7	8F	1F	3F	7F
7	FF	FF	FE	FC	F8	F1	E3	C7	8F	1F	3F	7F
8	FF	FF	FE	FC	F8	F1	E3	C7	8F	1F	3F	7F
9	FF	FF	FE	FC	F8	F1	E3	C7	8F	1F	3F	7F
10	FF	FF	FE	FC	F8	F1	E3	C7	8F	1F	3F	7F
11	FF	FF	FE	FC	F8	F1	E3	C7	8F	1F	3F	7F

The matrix PMASK′ is now updated as follows:

PMASK[i][j]=PMASK′[i][j] | (((((1<<i)-1)<<j)>>(i-1)) & PMASK′[0][j])
Where 0 < i < ws (number of numerical alphabets)
0 <= j < W-i+1
Final mask matrix PMASK is shown in Table 4.

4.2 Shift Value

Shift value for characters from fixed alphabet Σ is calculated as in BLIM algorithm. For the characters from parameterized set Π, the shift value for all the characters are same and is minimum of the shift values among them. For example, for the pattern P = abcba on Σ = {c} and Π = {a, b}

Character	Shift Value
a	8
b	8
c	10

4.3 Searching

Scan order in PBLIM algorithm is same as that of BLIM [14] algorithm. Firstly we compute a mask matrix MASK for the non-parameterized alphabets and an intermediate matrix PMASK' for the parameterized alphabet. We update the matrix PMASK' using the formulae given in Example 1 to get the mask matrix (PMASK) for parameterized alphabets. Now, we compute the shift value, scan order and *prev* encoding of the pattern as described above. In the matching algorithm, we store the value of the mask corresponding to the first scan value and store it in flag, then we logically AND the mask values corresponding to subsequent scan values in accordance with whether it is parameterized alphabet or non parameterized i.e. we logically AND it with the mask value of MASK if we get a non-parameterized set and we logically AND it with mask value of PMASK if it is in parameterized set. Then we check the value of flag. There is a match at those positions at which a bit is set in the flag. Finally, we move the window using the shift value of character immediately right to the window. The pseudo code of search algorithm is shown in Algorithm 1.

Algorithm 1. PBLIM (P, m, T, n)

```
Compute_MASK_Matrix(P, m)
Compute_PMASK_Matrix(P, m)
Compute_Shift_Value(P, m)
Compute_Scan_Order(m)
Compute_prev_Encode(P)
T′← Compute_prev_Encode(T)
k ← i ← 0
while i < n
   do if (T[i + ScanOrder[0]]∈ ∑)
        then flag←MASK[T′[i + ScanOrder[0]]][ScanOrder[0]]
      else flag←PMASK[T′[i + ScanOrder[0]]][ScanOrder[0]]
      for j ← 1 to ws - 1
        do if (T[i + ScanOrder[0]] ∈ ∑)
             then flag←Mask[T′[i+ScanOrder[j]]][ScanOrder[j]]
```

else flag←PMask[T′[i+ScanOrder[j]]][ScanOrder[j]];
if flag = 0^W
then break
if flag ≠ 0^W
then for j ← 0 to W - 1
do if (flag & $0^{W-j-1}10^j$)
then Pattern detected beginning at T[i + j]
i ←i+Shift[T[i + ws]]

4.4 Extension for Multiple Patterns

The PBLIM algorithm can be further extended to handle multiple patterns P_0, P_1, $P_2 \ldots P_{r-1}$ simultaneously. The extended algorithm is called as PMBLIM. In the preprocessing phase, the algorithm firstly calculates the *prev-encoding* [3] of all the patterns to be matched and subsequently keep on calculating the prev-encode of each *ws* length window of the text, where ws=Z+m_{max}-1, Z = $\lfloor w/r \rfloor$ and m_{max} is length of longest pattern. Here we use two mask matrices in the preprocessing phase: MMASK and PMMASK. The MMASK matrix containing the entries for fixed alphabets (Σ), is computed as in multi-pattern case of BLIM algorithm [14]. In this matrix we need not compute the mask values for characters that are in the parameterized set: Π. The second matrix PMMASK contains mask values of numbers from 0 to Z-1.

The mask matrix MMASK is computed using the BLIM algorithm [14]. This matrix contains entries for all the characters from fixed alphabet Σ but it need not contain the entries for characters that are in parameterized set:Π. An intermediate mask matrix PMMASK' is computed using the algorithm of multi-pattern cases of BLIM [14]. It contains entries for all the numeric alphabets (as in PBLIM). Now, this mask matrix PMMASK' is updated as follows:

Every Z bits of each mask value need to be updated using the formula discussed in the section 4.1. The matrix PMMASK′ is now updated as follows:

PMMASK[i][j][k…k+Z] = PMMASK′[i][j][k…k+Z] | (((((1<<i)-1)<<j)>>(i-1)) & PMMASK′[0][j][k…k+Z])

Where 0 < i < Z (number of numerical alphabets), 0 ≤ j < Z-i+1, 0≤k<r and PMMASK[i][j][k…k+Z] represents the k to k+Z bits, the mask value at i^{th} row and j^{th} column of the PMMASK'. The updated mask matrix PMMASK and MMASK are used in the main search loop.

The shift values for the characters not in parameterized set: Π, are calculated in the same way as in algorithm of multi-pattern cases of BLIM algorithm [14] and for characters in parameterized set the shift value is Z.

Scan order is computed using the same algorithm as discussed in multi-pattern section of BLIM algorithm [14]. In last, the main search loop used for matching is exactly identical to that discussed in the section 4.3 of PBLIM.

5 Experimental Results

We have implemented our proposed algorithm (PBLIM) and existing algorithm PSO [6] in C++, compiled with GCC 4.2.4 compilers on the Pentium 4, 2.14 GHz

Table 5. Comparison of time (in seconds) between PSO and PBLIM

Parameterized set	Pattern length (m)	Time (s) in **PSO**	Time (s) in **PBLIM**
{a }	10	5.097	5.024
	20	5.271	5.500
	30	5.823	6.000
	40	9.734	5.984
	50	9.379	6.880
{a, t }	10	5.271	5.004
	20	5.529	5.448
	30	5.932	5.988
	40	9.270	5.932
	50	9.867	6.720
{a, t, c }	10	5.467	4.972
	20	5.924	5.364
	30	5.883	5.960
	40	9.892	5.956
	50	10.027	6.724
{a, t, c, g}	10	5.349	5.004
	20	5.672	5.504
	30	5.982	6.148
	40	10.108	5.992
	50	10.114	6.824

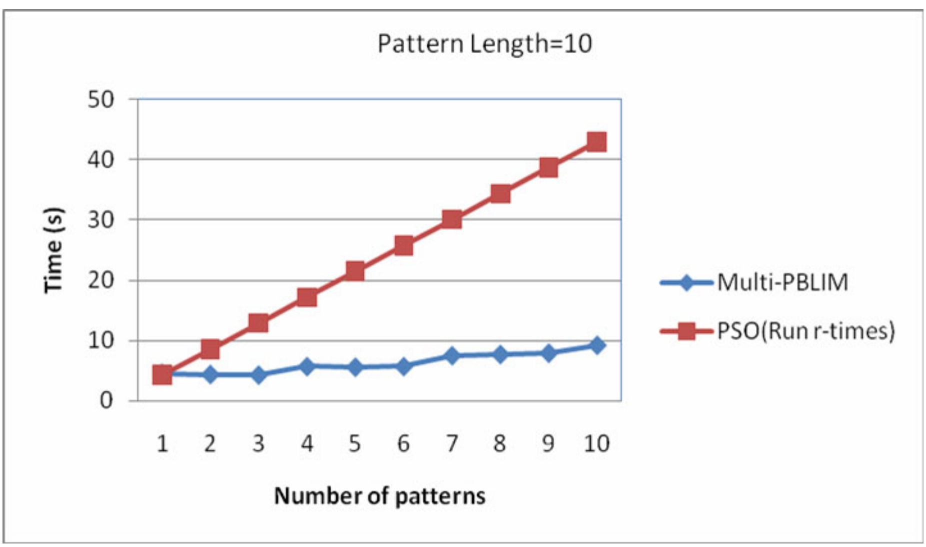


Fig. 1. Comparison between Shift-or (PSO) and PBLIM for multiple pattern case (pattern length = 10 is fixed)

processor (word length w= 32) with 512 MB RAM, running ubuntu 10.04 with file size 40 Mb from the file ftp://ftp.ncbi.nih.gov/genomes/H_sapiens/other/ of DNA alphabets. The patterns and text are chosen from the set {A, C, G, T}. Table 5 shows the comparison (of running time in seconds) between PSO and PBLIM algorithms. For $m > w$, we have taken an array of multiple words to implement the PSO. Fig. 1 shows the comparison between parameterized shift-or (PSO) and PBLIM algorithm for multiple patterns cases.

6 Conclusions and Future Work

In this paper, we have extended the BLIM algorithm of exact string matching for parameterized string matching. We compare the proposed algorithm (PBLIM) with parameterized shift-or (PSO) [6] algorithm for the cases: (i) when the pattern length is within the word size (w), (ii) when the pattern length is greater than w and (iii) for multiple pattern cases. For the first case, from Table 5, it is clear that our proposed algorithm PBLIM is comparable with PSO. For the second case, it is substantially better than PSO. From Table 5, it is also clear that, on increasing the size of parameterized set (i.e. on increasing the duplicity in the text), time increases. Hence as the number of clones in a software system increases, it is easier to detect clones by PBLIM algorithm as compared to PSO. Multiple patterns BLIM can be further compared with multiple patterns shift-or, where multi-patterns in shift-or is being handled by either classes of characters or by method of concatenation [15].

References

1. Roy, C.K., Cordy, J.R.: A survey on clone detection research. Technical Report No. 2007-541, School of Computing, Queen's University at Kingston, Ontario, Canada (2007)
2. Baeza-Yates, R.A., Gonnet, G.H.: A new approach to text searching. Communication of ACM 35(10), 74–82 (1992)
3. Baker, B.S.: Parameterized duplication in string: algorithm and application in software maintenance. SIAM J. Computing 26(5), 1343–1362 (1997)
4. Baker, B.S.: Parameterized diff. In: 10th Symposium on Discrete Algorithm (SODA), pp. 854–855 (1999)
5. Boyer, R.S., Moore, J.S.: A fast string-searching algorithm. Communication of ACM 20(10), 762–772 (1977)
6. Fredriksson, K., Mozgovoy, M.: Efficient parameterized string matching. Information Processing Letters (IPL) 100(3), 91–96 (2006)
7. Horspool, R.N.: Practical fast searching in strings, Software. Practice & Experience 10(6), 501–506 (1980)
8. Prasad, R., Agarwal, S.: A new parameterized string matching algorithm by combining bit-parallelism and suffix automata. In: 8th IEEE International Conference on Computer and Information Technology, Sydney, Australia, pp. 778–783. IEEE Press, Los Alamitos (2008)
9. Raita, T.: Tuning the Boyer-Moore-Horspool string searching algorithm. Software - Practice & Experience 22(10), 879–884 (1992)

10. Salmela, L., Tarhio, J.: Fast Parameterized Matching with q-grams. Journal of Discrete Algorithm 6(3), 408–419 (2008)
11. Smith, P.D.: Experiments with a very fast substring search algorithm. Software - Practice & Experience 21(10), 1065–1074 (1991)
12. Sunday, D.M.: A very fast substring search algorithm. Communications of the ACM 33(8), 132–142 (1990)
13. Wu, S., Manber, U.: Fast text searching allowing errors. Communication of the ACM 35(10), 83–91 (1992)
14. Kulekci, M.O.: BLIM: A New Bit-Parallel Pattern Matching Algorithm Overcoming Computer Word Size Limitation. Mathematics in Computer Science 3(4), 407–420 (2010)
15. Navarro, G., Raffinot, M.: Fast and Flexible String Matching by Combining Bit-parallelism and Suffix automata. ACM Journal of Experimental Algorithms 5(4) (2000)

Behavior Analyzer for Developing Multiagent System on Repository-Based Multiagent Framework

Takahiro Uchiya[1] and Tetsuo Kinoshita[2]

[1] Nagoya Institute of Technology, [2] Tohoku University
[1] Gokiso-chou, Showa-ku, Nagoya, 466-8555 Japan
[3] 2-1-1 Katahira, Aoba-ku, Sendai, 980-8577 Japan
t-uchiya@nitech.ac.jp, kino@riec.tohoku.ac.jp

Abstract. Agent systems have been designed and developed using recent agent technologies. However, design and debugging of these systems remain as difficult tasks of designers because agents have situational and nondeterministic characteristics and because any useful design support facilities have not been provided for designers. To raise the efficiency of the agent system design process, we propose an interactive design environment of agent system (IDEA) founded on an agent-repository-based multiagent framework. In this paper, we focus on the function of behavior analyzer for developing multiagent system and show the effectiveness of this function.

Keywords: interactive design; multiagent framework; design environment.

1 Introduction

Software with new characteristics such as autonomy and sociality is called an agent; an information system that uses agents as its components is called an agent system. Agent systems of many kinds have been designed and developed using recent agent technologies for Internet Auction, Network Management System, Video Conference, Smart Grid, etc.

However, the design and debugging of agent systems persists as a difficult problem not only because of the situational and nondeterministic behavioral properties of the agents, but also because of the lack of an effective design method and design-support technologies. To date, we have studied an agent-repository-based multiagent framework called ADIPS, which accumulates the developed agents and agent systems in an agent repository and which enables the dynamic composition and re-composition of agent systems based on this repository [1][2]. Applying the ADIPS/DASH framework, a recent implementation of ADIPS framework with an effective agent repository [3][4][5], we developed various agent systems in our previous work [6][7][8][9] [10][11][12][13]. Through the experience of developing many agent systems, we paused to realize the importance of the design support tools to raise the efficiency of the agent system design process.

So, in our research, we propose an Interactive Design Environment of Agent system (IDEA) founded on an agent-repository-based multiagent framework. In this

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 410–420, 2011.

paper, we especially focus on the function of behavior analyzer for designing and developing multiagent system and show the effectiveness of this function.

2 Problems in Agent System Design

2.1 Design Task of Agent System

The following two approaches can be considered from a viewpoint of implementation of the agent system: (i) a Programming Approach and (ii) a Framework Approach.

(i) Programming Approach:
In this approach, an agent is designed and implemented in a top-down way using existing programming languages such as Java. Design flexibility can be retained in the design process. Therefore, agents with dedicated architecture like basic-type and the reactive-type agents can be implemented easily. However, for deliberative-type and composite-type agents, the burden of designers is expected to be increased because of the design and implementation of the internal mechanisms embedded in all agents.

(ii) Framework Approach:
By providing an agent design support environment based on the specific agent architecture for the designers, the design and implementation of the agents can be done systematically and efficiently. Such an environment, a “framework”, provides facilities such as a knowledge representation scheme, a problem-solving function, and the agent communication function for the designers. In recent years, many frameworks such as ADIPS [1][2], JADE [14], SAGE [15], AgentBuilder [16], JATLite [17], ZEUS [18], and JACK [19] have been proposed and used. These frameworks might provide many functions to design and implement various agents for designers. These frameworks typically provide user-oriented support facilities for design, implementation, debugging, and testing.

2.2 Problems of Agent System Design

From the perspective of a state-of-the art of agent system design, we emphasize two problems.

(P1) The designer’s burden might increase in direct relation to the size of the target agent system to be designed. For that reason, systematic bottom-up design processing, by which the developed agents are reused, should be supported.
(P2) The costs of both testing and debugging of the agent system become larger than that of the non-agent system, due to non-deterministic behavior of agent systems. Design support functions by which designers can interact with the target agents flexibly should be required.

To overcome these problems, firstly we use the repository-based multiagent framework to solve (P1). Moreover, we propose a prototype of the design support environment called IDEA to solve (P2).

3 Repository-Based Multiagent Framework

3.1 Basic Concept

The ADIPS/DASH framework is a latest repository-based multiagent framework developed by our research group (Fig. 1). The essential functions of this framework can be summarized as follows.

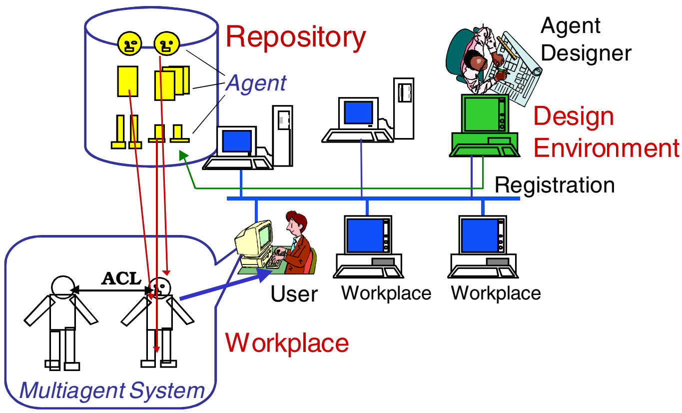


Fig. 1. ADIPS/DASH: Repository-based multiagent framework

F1. Repository-based multiagent framework for distributed problem solving
ADIPS/DASH framework provides a multiagent platform over the distributed environment, which consists of the distributed Agent Workplaces (abbreviated as the workplaces) and the Agent Repository (abbreviated as the repository).

The repository manages various DASH agents (abbreviated as the agents) and is responsible to design and realize the multiagent systems based on the users' requests.

A workplace is an agent execution environment on a distributed computer platform and is responsible to monitor and control the behavior of agents realized by the repository.

The user can design various multiagent systems by sending his/her request to the repository and realize the system on the workplace to execute the problem solving tasks. Hence, this framework is called the Repository-based multiagent framework.

F2. Building a multiagent system by Agent Repository
A multiagent system is constructed in the repository and delivered to the workplace. The organization procedure in the agent repository (Fig. 2) is basically as same as the Contract Net protocol [20], however, the following functions such as the instantiation of organization onto the workplace and the reconstruction of organization at the run time of agent systems, are unique and important features of ADIPS/DASH framework as the developing and runtime environment of agent systems.

A user and/or an agent run on the workplace can send a request to the repository to start a design process of a multiagent system, which provides a required service for the user/agent. Receiving a request, the repository tries to find out suitable agents and construct an agent organization (a multiagent system) using the Extended Contract Net Protocol (ECNP) of ADIPS/DASH framework.

The ECNP is an agent cooperation protocol to deal with the design processes of construct, reconstruct and instantiate the multiagent systems.

When a required agent organization is successfully constructed in the repository, the repository instantiates the agent organization as a multiagent system run on the specified workplaces. Hence, the multiagent system can be realized with respect to the given request in a dynamic way.

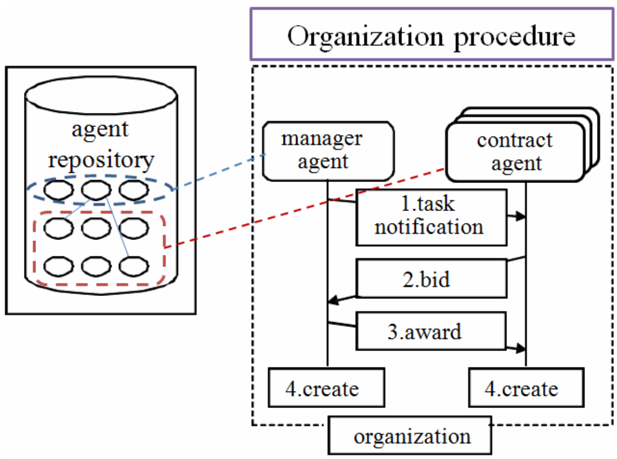


Fig. 2. Organization procedure in the repository on the ADIPS/DASH framework

F3. Using multiagent systems on Agent Workplace

When a multiagent system is instantiated on a workplace by the repository, the workplace activates this system to start a problem solving task. The workplace monitors the behavior of these agents and collects runtime information such as execution log.

When the problem solving task of the multiagent system is finished, the agent organization is dissolved by the workplace. All agents of the multiagent system are stopped and removed from the workplace.

On the other hand, it is possible to construct a multiagent system which consists of the active agents run on the distributed workplaces by using ECNP. In this case, a construction process based on ECNP is similar to the original CNP. Moreover, in order to use/reuse the realized multiagent system in the future, the workplace can save the multiagent system in the repository by a request for preservation.

F4. Reconstruction of multiagent system at the runtime

It is possible to reconstruct the structure and functions of a multiagent system at the runtime of the system based on a request of a user or a problematic agent in the multiagent system. The ECNP provides the functions and protocols for reorganization operation.

F5. Rule-based agent programming

The design of an agent is to describe the behavior knowledge for cooperative problem solving together with the meta knowledge for managing the agent in the repository.

The behavior knowledge is described as a set of rules using the rule-type knowledge representation format, on the other hand, the meta knowledge is described using frame-type knowledge representation format. The description of the designed agent is called the agent program, which are interpreted and executed using an inference engine (production system type engine) of DASH agent.

F6. "Rule Set" for reusable knowledge

The general-purpose and/or useful behavior knowledge can be defined as a predefined set of rules, called "Rule Set" and stored in the repository, in order to support the agent design task based on the use/reuse of the rule sets.

The agent designer can select and specify the suitable rule sets in an agent program. In the repository, the specified rule sets are included in a knowledge base of the agent. A set of rules for handling a task oriented cooperation protocol is an example of the rule set.

F7. Wrapping external program

The ADIPS/DASH framework provides a wrapping mechanism for the agent designers to utilize the external programs such as Java programs as the procedural knowledge of the agent.

F8. Interoperation with other agents

The interoperation mechanism can be included in the ADIPS/DASH framework. Using this mechanism, the DASH agents can communicate with the different type of agents such as FIPA-compliant agents using the ACL messages of DASH agent [21].

F9. Test, debug and validation of multiagent system

The Interactive Design Environment of Agent system (IDEA) provides a design environment of agent/multiagent system. The details of IDEA are appeared in next section.

4 Prototype of Interactive Design Environment of Agent System

We developed a prototype of an interactive design support system called the Interactive Design Environment of Agent system (IDEA) using Java program (Sun Java SE 6). We select and use a repository-based multiagent framework, ADIPS/DASH, for implementation of the IDEA prototype. The functional relations between the ADIPS/DASH framework and IDEA are depicted in Fig. 3.

The ADIPS/DASH is used to realize an agent execution environment equipped with the repository mechanism. In addition, IDEA provides an interactive design environment for designers of agent systems.

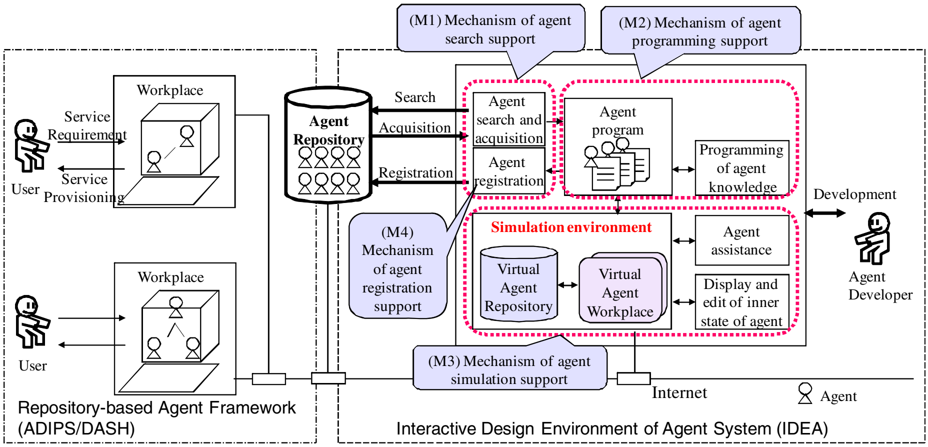


Fig. 3. Overview of Interactive Design Environment of Agent system (IDEA)

The following four mechanisms are introduced into IDEA to support agent design.

<u>(M1) Mechanism of agent search support</u>
Its three main parts are a search condition input for seeking agents from the repository, a search result display, and a preview of the agent knowledge.

In the search condition input area, a designer inputs the requirement specification of a candidate agent, such as an agent name and a function name. The received message is displayed in the search result display area when a candidate agent, which is detected in the repository, sends a message as its response. The designer can then move an agent into the developer's environment by choosing the agent from a search result window.

<u>(M2) Mechanism of agent programming support</u>
This mechanism has an agent-programming editor based on a rule-based knowledge representation of the ADIPS/DASH framework. Using this editor, the designer can describe and test the agent programs.

<u>(M3) Mechanism of agent simulation support</u>
In this paper, we focus on the function of behavior analyzer for designing and developing multiagent system. This mechanism has some interactive simulation functions to analyze agent's behavior such as "virtual distributed environment", "exchange messages between designer and agents", "dynamic knowledge modification", and "message log analyzer". Fig. 4 shows an agent monitor for observing the behavior and organization of the agent in the virtual distributed environment: one repository and five workplaces. The agent inspector shows the inner states of an agent to monitor and modify the behavioral knowledge of the runtime agent. Furthermore, the ACL editor supports communication between the designer and the agents under development by the ACL messages of the ADIPS/DASH framework.

Using these tools, the designer can monitor and control the behavior of agents in an interactive manner during the testing and debugging of agents. Moreover, the designer can access other design stages at any time by selecting the design stage tabs shown at the upper part of screen. For instance, by selecting the "Design" tab, the designer goes to the "Design" stage to modify the agent program; then the designer will again view the "Simulation" stage and resume the simulation by reloading the modified agents.

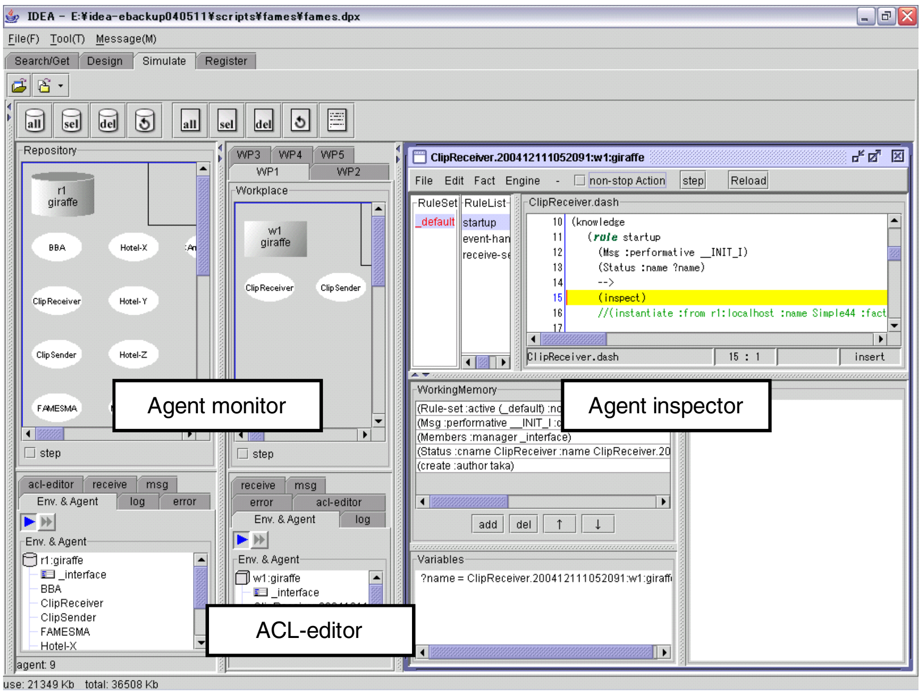


Fig. 4. Overview of behavior analyzer

<u>(M4) Mechanism of agent registration support</u>
This mechanism provides an interface to store the completed agent system to the repository.

5 Experiments and Evaluation

5.1 Experiments

To verify IDEA's effectiveness, we tested the effects of behavior analysis functions: "virtual distributed environment", "exchange messages between designer and agents", "dynamic knowledge modification", and "message log analyzer".
[Result of the function of virtual distributed environment]

The agent monitor of the simulation interface showed dynamic behavior of distributed agents in real time; the designers can observe non-deterministic behavior such as organization, communication, cooperation, and competition of agents on the repository/workplaces. Therefore we confirmed this function is useful to develop the distributed agent system.

[Result of the function of exchange messages between designer and agents]
The ACL-editor tab (Fig. 5) enables the designer to send ACL-messages to agents under development. Consequently, the designer can select and run an agent that is a part of the agent system to test and verify the functions of the agent and the agent system smoothly.

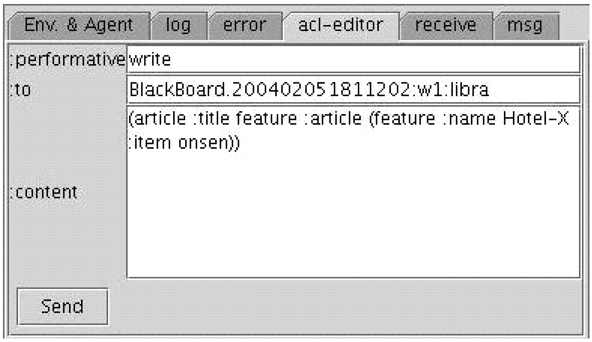


Fig. 5. ACL-editor tab

[Result of the function of dynamic knowledge modification]
The agent inspector shown in Fig. 4 enables the designer to modify knowledge of agents dynamically at the runtime of an agent system.

Thereby, the designer can test and modify the functions of the agent system both quickly and smoothly.

[Result of the function of message log analyzer]
The message log analyzer is portrayed in Fig. 6 and Fig. 7.

Fig. 6 shows the agent map for analyzing message flow of the whole agent system.

Fig. 7 shows the messaging monitor for analyzing the message sequence of the agent system. They are generated using behavioral logs of agents stored in IDEA.

The agent developer could detect a bottleneck that restricts message passing of agent systems by using this tool. Moreover developer could determine the causes of agents' abnormal behavior by monitoring message sequence.

[Overall evaluation]
We confirmed that all functions are effective for developing agent system.

5.2 Comparison with Related Framework

JADE is an extremely well-known agent framework used to develop Java-based multiagent systems. In terms of behavior analysis of agent systems, we can compare IDEA with JADE.

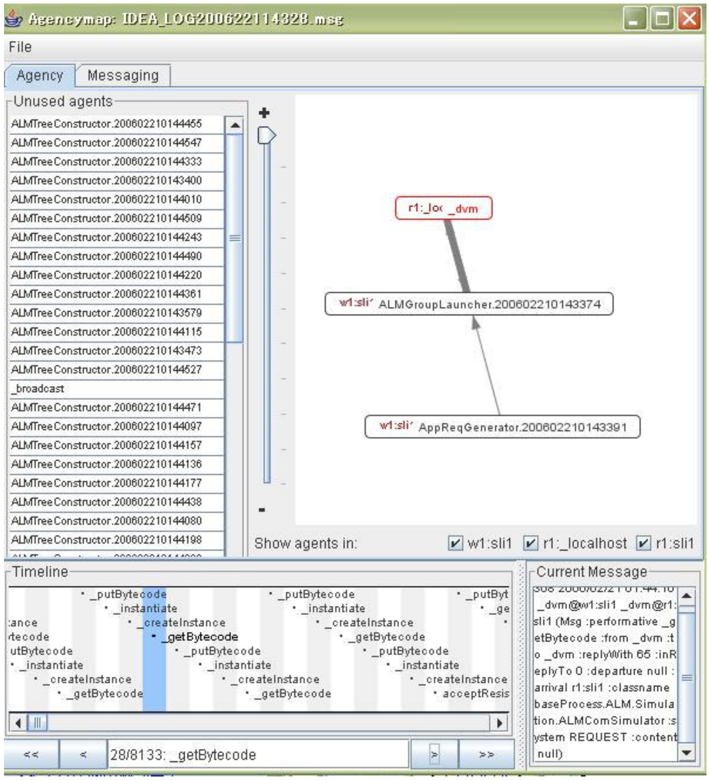


Fig. 6. Agent map for analyzing message flow of the whole agent system

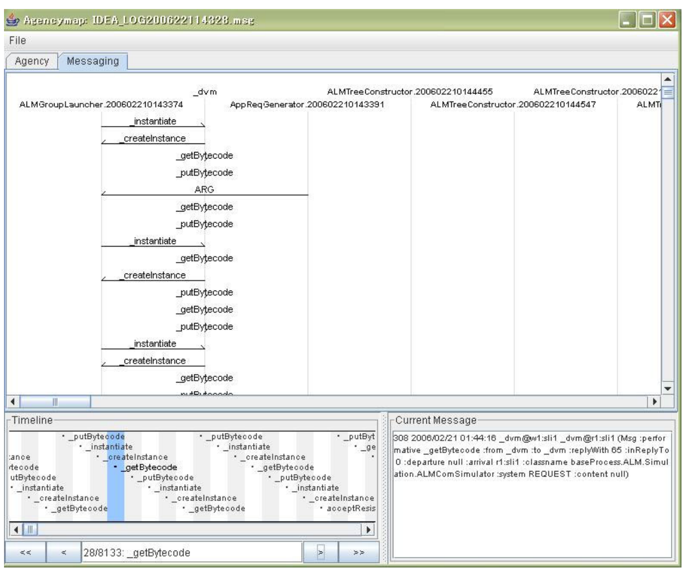


Fig. 7. Messaging monitor for analyzing message sequence of the agent system

[Mutual functions]

<1. Virtual distributed environment> JADE framework has some agent working environment called “container” and can simulate and monitor the agent behavior in virtual distributed environment.

<2. Exchange messages between designer and agents> JADE framework supports the message exchange between designer and agents by sniffer function. However, the specifiable message parameter is restricted. IDEA can set the message parameter: “performative (communicative act)”, “to” and “content”, but JADE can set the message parameter: only “to” and “content”.

<3.Messaging monitor for analyzing message sequence> JADE’s sniffer provides this function.

[Unique functions]

<4. Dynamic knowledge modification”> IDEA can change the agent knowledge dynamically at run-time. This function helps us to test and debug the unstable agents or uncompleted developed agents.

<5. Agent map for analyzing message flow> IDEA can monitor the message flow and detect the bottle-neck agent or busy agent. This function is indispensable for simulation of massive agent system.

6 Conclusion

In this paper, we proposed a design support facility of agent systems based on the repository-based multiagent framework to provide an efficient and systematic design environment for agent system designers. A prototype of the IDEA is implemented.

Experimental results demonstrate the effectiveness of the IDEA in the development of agent systems. In future works, we will enhance the analysis function of agent behavior to test and debug the agent system more smoothly. Moreover, we will expand the proposed design environment by accumulating numerous design cases of agent-based intelligent applications.

References

1. Kinoshita, T., Sugawara, K.: ADIPS framework for flexible distributed systems. In: Ishida, T. (ed.) PRIMA 1998. LNCS (LNAI), vol. 1599, pp. 161–175. Springer, Heidelberg (1999)
2. Fujita, S., Hara, H., Sugawara, K., Kinoshita, T., Shiratori, N.: Agent-based design model of adaptive distributed system. Applied Intelligence 9(1), 57–70 (1998)
3. Hara, H., Sugawara, K., Kinoshita, T., Uchiya, T.: Flexible distributed agent system and its application. In: Proceedings of the Fifth Joint Conference of Knowledge-Based Software Engineering, pp. 72–77 (2002)
4. Uchiya, T., Suganuma, T., Kinoshita, T., Shiratori, N.: An architecture of active agent repository for dynamic networking. In: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 1266–1267 (2002)
5. Uchiya, T., Takeda, A., Suganuma, T., Kinoshita, T., Shiratori, N.: A method for realizing user-oriented service with repository-based agent framework. In: Proceedings of the 1st Int. Forum on Information and Computer Technology (IFICT 2003), IPSJ, pp. 119–124 (2003)

6. Suganuma, T., Imai, S., Kinoshita, T., Sugawara, K., Shiratori, N.: A flexible videoconference system based on multiagent framework. IEEE Trans. on Systems, Man, and Cybernetics – Part A: Systems and Humans 33(5), 633–641 (2003)
7. Imai, S., Kitagata, G., Konno, S., Suganuma, T., Kinoshita, T.: Developing a knowledge-based videoconference system for non-expert users. Journal of Distance Education Technologies 2(2), 13–26 (2004)
8. Konno, S., Iwaya, Y., Abe, T., Kinoshita, T.: Design of network management support system based on active information resource. In: Proceedings of the 18th International Conference on Advanced Information Networking and Application, pp. 102–106 (2004)
9. Kitagata, G., Matsushima, Y., Hasegawa, D., Kinoshita, T., Shiratori, N.: An agent-based middleware for communication service on ad hoc network. In: Proceedings of the 19th Int. Conference on Advanced Information Networking and Application, pp. 363–367 (2005)
10. Takahashi, A., Suganuma, T., Abe, T., Kinoshita, T.: Dynamic construction scheme of multimedia processing system based on multiagent framework. The International Journal of Wireless and Mobile Computing 2(1) (2006)
11. Abar, S., Konno, S., Kinoshita, T.: Autonomous network monitoring system based on agent-mediated network information. The International Journal of Computer Science and Network Security 8(2), 326–333 (2008)
12. Takahashi, H., Izumi, S., Suganuma, T., Kinoshita, T., Shiratori, N.: Multi-agent system for user-oriented healthcare support. The International Journal of Informatic Society (IJIS) 1(3), 32–41 (2009)
13. Kim, H., Kinoshita, T., Lim, Y., Kim, T.: A bankruptcy problem approach to load-shedding in multiagent-based microgrid operation. Sensors 10(10), 8888–8898 (2010)
14. Bellifemine, F., Poggi, A., Rimassa, G.: JADE – a FIPA-compliant agent framework. In: Proc. of Practical Application of Intelligent Agents and Multi Agents, pp. 97–108 (1999)
15. Ghafoor, A., ur Rehman, M., Khan, Z.A., Ali, A., Ahmad, H.F., Suguri, H.: SAGE: next generation multi-agent system. In: Proceedings of Parallel and Distributed Processing Techniques and Applications, pp. 139–145 (2004)
16. Reticular Systems. AgentBuilder – An integrated toolkit for constructing intelligence software agents, http://www.agentbuilder.com/
17. Jeon, H., Petrie, C., Cutkosky, M.: JATLite: a java agent infrastructure with message routing. IEEE Internet Computing 4(2), 87–96 (2000)
18. Nwana, H.S., Ndumu, D.T., Lee, L.C., Collins, J.C.: ZEUS: a toolkit for building distributed multi-agent systems. Applied Artificial Intelligence Journal 13(1), 129–186 (1999)
19. Renquist, N.R., Andrew, H., Andrew, L.: Jack – summary of an agent infrastructure. In: Proceedings of 5th International Conference on Autonomous Agents (2001)
20. Smith, R.G.: The contract net protocol: high-level communication and control in a distributed problem solver. IEEE Trans. on Computers 29(12), 1104–1113 (1980)
21. Li, X., Uchiya, T., Konno, S., Kinoshita, T.: Proposal for agent platform dynamic interoperation facilitating mechanism. In: Nguyen, N.T., Borzemski, L., Grzech, A., Ali, M. (eds.) IEA/AIE 2008. LNCS (LNAI), vol. 5027, pp. 825–834. Springer, Heidelberg (2008)

Impact Analysis of Goal-Oriented Requirements in Web Engineering

José Alfonso Aguilar[1,2], Irene Garrigós[1], and Jose-Norberto Mazón[1]

[1] Department of Software and Computing Systems
University of Alicante, Spain
[2] Computer Science Faculty
University of Sinaloa, Mexico
{ja.aguilar,igarrigos,jnmazon}@dlsi.ua.es

Abstract. Due to the continuous changes and heterogeneous audience of the Web, a requirement engineering stage is crucial for Web development. Importantly, this stage should consider that Web applications are more likely to rapidly evolve during the development process, thus leading to inconsistencies among requirements. Therefore, Web developers need to know dependencies among requirements to ensure that Web applications finally satisfy the audience. The understanding of requirement dependencies also helps in better managing and maintaining Web applications. In this work, an algorithm has been defined in order to deal with dependencies among functional and non-functional requirements to understand which is the impact of making changes when developing a Web application.

Keywords: Impact analysis, goal-oriented requirements engineering, Web engineering.

1 Introduction

Requirements in Web engineering tend to rapidly evolve due to the dynamic nature of the Web [1]. This continuous evolution may lead to inconsistencies among requirements which hinder Web developers from understanding the impact of a change in the Web application. To tackle this problem, dependencies among requirements should be explicitly considered, thus better managing and maintaining requirements in Web applications. Dependencies are essential elements among the requirements of a real software system to achieve certain stakeholder goals. On the other hand, inconsistencies are the negative dependencies among the set of requirements, caused by the fact that requirements often originate from stakeholders with different or conflicting viewpoints [2].

Impact analysis is the task of identifying the potential consequences of a change, or estimating what needs to be modified to accomplish a change [3]. We define a "change" as any modification on a Web requirement. Usually, impact analysis has been done intuitively by Web applications developers, after some cursory examination of the code and documentation. This may be sufficient for

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 421–436, 2011.

small Web applications, but it is not enough for sophisticated ones. In addition, empirical investigation shows that even experienced Web applications developers predict incomplete sets of change impacts [4].

Therefore, to effectively manage changes in Web applications, information about dependencies among requirements should be considered in order to know how changes in a Web requirement affect the other requirements, thus making the right design decisions.

Usually, impact analysis is performed only on functional requirements, leaving aside the non-functional requirements as shown in [5,6]. According to [7], we believe that non-functional requirements must be considered in the Web application development from the very beginning of the development process. It is worth noting that the impact analysis should be done on both kind of requirements: functional and non-functional.

Functional requirements describe system services, behavior or functions, whereas non-functional requirements, or quality requirements, specify a constraint on the system or on the development process [8]. Functional requirements are related to *goals* and *sub-goals* in goal-oriented modeling. Non-functional requirements are named *softgoals* in goal-oriented modeling to represent objectives that miss clear-cut criteria. Specifically, finding the right tradeoff for non-functional requirements is an important step for achieving successful software [9,10], i.e., a Web application without passwords is usable, but not very secure, increased usability reduce security or increased security reduce usability. Therefore, it is necessary to identify the dependencies among non-functional and functional requirements for their achievement. Unfortunately, finding this tradeoff is not a trivial task due to the conflicts that commonly arise among them. Interestingly, the recent inclusion of goal-oriented techniques in Web requirements engineering [11,12,13,14] offer a better analysis in Web application design, due to the fact that requirements are explicitly specified in goal-oriented models. This has allowed the *stakeholders* to understand among the design decisions that can be taken to satisfy their goals and evaluating the implementation of certain functional and non-functional requirements in particular. However, this is not enough to ensure that the Web application satisfies the goals.

This paper presents a goal-oriented proposal for supporting Web developers analyzing the impact of a requirement change in Web applications in order to provide information about the different design alternatives. Therefore, more informed design decisions can be made for developing a Web application that fully-satisfies goals, while there is a tradeoff among softgoals.

The remainder of this paper is structured as follows: Section 2 presents some related work relevant to the context of this work. Section 3 describes the proposal for goal-oriented requirements analysis where is found the contribution of this work and introduces a running example for demonstration purposes. The algorithm for impact analysis in goal-oriented requirements is presented in Section 4. The application of the algorithm to perform the impact analysis is described step by step in Section 5. Finally, the conclusion and future work is presented in Section 6.

2 Related Work

In our previous work [15], a systematic literature review has been conducted for studying requirement engineering techniques in the development of Web applications. Our findings showed that most of Web engineering approaches focus on the analysis and design phases and do not give a comprehensive support to the requirements phase (such as OOHDM [16], WSDM [17] or Hera [18]). We can also conclude that the most used requirement analysis technique is UML use cases (applied by OOWS [19], WebML [20], NDT [21] and UWE [22]).

Furthermore, none of the aforementioned Web engineering approaches perform the analysis and modeling of the users' needs for ensuring that the Web application satisfies real goals, i.e. users are not overloaded with useless functionalities, while important functionalities are not missed. We believe that these facts are an important issue that limits a broaden use of these approaches. In this sense, to the best of our knowledge, the only approaches that use goal-oriented requirements analysis techniques for Web engineering have been presented in [23,24]. Unfortunately, although these approaches use the i^* modeling framework [25,26] to represent requirements in Web domain, they do not benefit from every i^* feature. To overcome this situation, our previous work [13] adapts the well-known taxonomy of Web requirements presented in [27] for the i^* framework.

Regarding approaches that consider non-functional requirements from early stages of the development process, in [28] the authors propose a metamodel for representing usability requirements for Web applications. Moreover, in [7] the authors present the state-of-the-art for non-functional requirements in model-driven development, as well as an approach for integrating non-functional requirements into a model-driven development process by considering them from the very beginning of the development process. Unfortunately, these works overlook how to analyze and evaluate the impact among functional and non-functional requirements. However, some interesting works have been done in this area [29] and [30]. These works evaluate i^* models based upon an analysis question (what-if) and the human judgment. To this aim, this procedure uses a set of evaluation labels that represent the satisfaction or denial level of each element in the i^* model. First of all, initial evaluation labels reflecting an analysis question are placed in the model. These labels are then propagated throughout the model by using a combination of set propagation rules and the human judgment. The results of this propagation are interpreted in order to answer the stated question. Unfortunately, these general approaches have not been adapted to Web engineering.

The motivation regarding this proposal relies in the fact that, the works previously mentioned are focused on how to analyze i^* models to answer a particular question (what-if) without considering the goal satisfaction (the organizational objectives). Thus, our proposal is focused on how to evaluate the impact derived from a change in a i^* requirements model in the Web engineering field. For this purpose, in the i^* model the requirements are classified according the taxonomy of Web requirements defined in [27]. Also, our proposal brings out alternative paths to satisfy the goals bearing in mind the softgoals tradeoff, hence, considering the softgoals from the beginning of the Web application development process.

To sum up, there have been many attempts to provide techniques and methods to deal with some aspects of the requirements engineering process for Web applications. However, there is still a need for solutions that allow an impact analysis by considering non-functional requirements.

3 Goal-Oriented Requirements Analysis in Web Engineering

This section briefly describes our proposal to specify requirements in the context of a Web modeling method by using i^* models [14]. As a goal-oriented analysis technique, the i^* framework focuses on the description and evaluation of alternatives and their relationships to the organizational objectives. This proposal supports an automatic derivation of Web conceptual models from a requirements model by means of a set of transformation rules [31]. The proposal presented in this paper is defined upon our Web engineering method A-OOH *(Adaptive Object Oriented Hypermedia method)* [32] although it can be applied to any other Web engineering approach.

Next, we shortly describe an excerpt of the i^* framework which is relevant for the present work. For a further explanation, we refer the reader to [25,26]. The i^* framework consists of two models: the strategic dependency (SD) model to describe the dependency relationships (represented as) among various actors in an organizational context, and the strategic rationale (SR) model, used to describe actor interests and concerns and how they might be addressed. The SR model (represented as) provides a detailed way of modeling internal intentional elements and relationships of each actor (). Intentional elements are goals (), tasks (), resources () and softgoals (). Intentional relationships are means-end links () representing alternative ways for fulfilling goals; task-decomposition links () representing the necessary elements for a task to be performed; or contribution links ($\xrightarrow[\text{hurt}]{\text{help}}$) in order to model how an intentional element contributes to the satisfaction or fulfillment of a softgoal.

Although i^* provides good mechanisms to model actors and relationships between them, it needs to be adapted to the Web engineering domain to reflect special Web requirements that are not taken into account in traditional requirement analysis approaches. To do this, in first place, we use the taxonomy of Web requirements presented in [27]:

- **Content Requirements.** With this type of requirements, is defined the website content presented to users. For example, in a book on-line store some examples might be: “book information” or “book categories”.
- **Service Requirements.** This type of requirement refers to the internal functionality the system as Web application should provide to its users. Following the example of the Content Requirements, for instance: “register a new client”, “add book to cart”, etc.

– **Navigational Requirements.** A Web system must also define the navigational paths available for the existing users. In this sense, some examples are: the user navigation from index page to "consult products by category" or to "consult shopping cart" options.
– **Layout Requirements.** Requirements can also define the visual interface for the users. For instance: "present a color style", "multimedia support", "the user interaction", among others.
– **Personalization Requirements.** The designer can specify the desired personalization actions to be performed in the final website (e.g. "show recommendations based on interest", "adapt font for visual impaired users", etc.)
– **Non-Functional Requirements.** These kind of requirements are related to quality criteria that the intended Web system should achieve and that can be affected by other requirements. Some examples can be "good user experience", "attract more users", "efficiency", etc.

As the considered Web engineering approach (A-OOH) is UML-compliant, we have used the extension mechanisms of UML to (i) define a profile for using $i*$ within UML; and (ii) extend this profile in order to adapt $i*$ to specific Web domain terminology. Therefore, new stereotypes have been added according to the different kind of Web requirements (see Fig. 1): *Navigational*, *Service*, *Personalization* and *Layout* stereotypes extend the *Task* stereotype and *Content* stereotype extends the *Resource* stereotype. It is worth noting that non-functional requirements can be modeled by directly using the softgoal stereotype. The UML-Profile for goal-oriented requirements using $i*$ has been implemented in Eclipse [33].

A sample application of the $i*$ modeling framework for Web domain is shown in Figure 2, which represents the SR model of our running example for the Conference Management System (CMS). The purpose of the system is to support

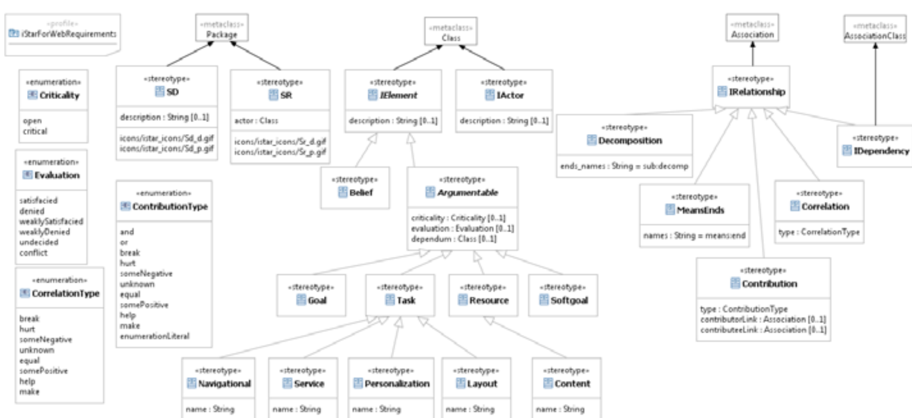


Fig. 1. Overview of the UML-Profile for $i*$ modeling in the Web domain implemented in Eclipse

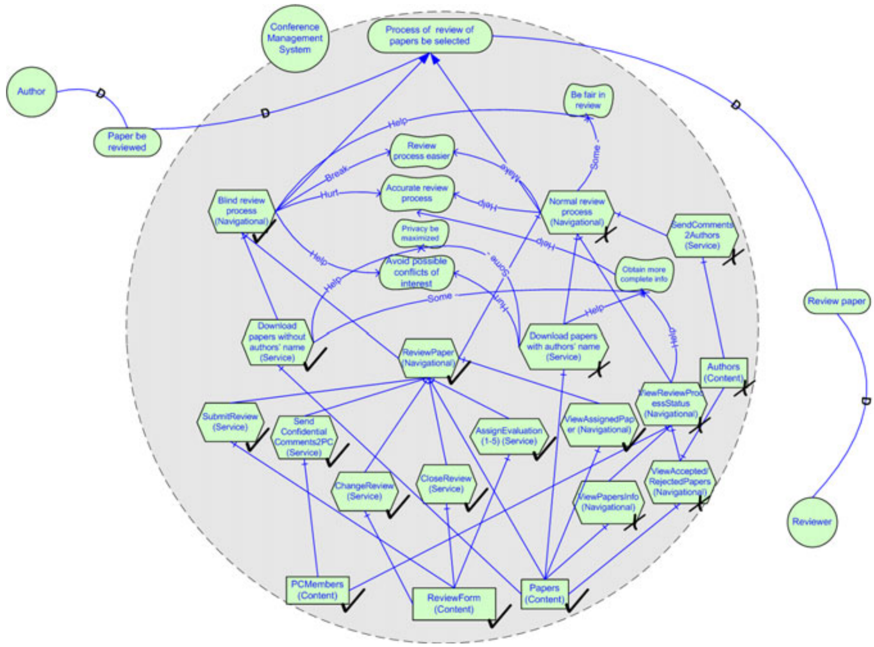


Fig. 2. Part of the Conference Management System requirements expressed in a SR and SD Models

the process of submission, evaluation and selection of papers for a conference [34].[1]

In Figure 2 is modeled a part of the CMS focused in the process of selecting the review process. Four actors participate in the CMS, but due to space limitations, for this example only the author, reviewer and system actors were considered. It is important to highlight that each element from Figure 2 corresponds to a requirements type from the taxonomy previously mentioned, i.e., the content requirement (Content) from the taxonomy is displayed with the notation *"Resource"* from *i** and the navigational (Navigational) and service (Service) requirements with the symbol *"Task"* from *i**, both with their respective associations (*decomposition-links*). A decomposition-link between two elements means that a requirement (*"Task"*) is decomposed in one or more sub-requirements (*"Sub-Tasks"*), Figure 2 depicts a correct scenario of the requirement decomposition by introducing the navigational requirement *"Blind Review Process"*, decomposed in two sub-requirements named *"Download papers without authors' name"* (Service) and *"Review Paper"* (Navigational). Also, the labels (✓) and (✗) are used to represent the requirements currently implemented in the Web application.

---

[1] The complete specification of the case study can be found at: `http://users.dsic.upv.es/~west/iwwost01`

Three actors are detected that depend on each other, namely *"Reviewer"*, *"Author"* and *"Conference Management System"*. The reviewer needs to use the CMS to *"Review paper"*. The author depends on the CMS in order to *"Paper be reviewed"*. These dependencies and the CMS actor are modeled by a SD and SR models in Figure 2.

The goal of the CMS actor is *"Process of review of papers be selected"*. To fullfill this goal, the SR model specifies that one of the two navigational requirements: *"Blind review process"* or *"Normal review process"* should be performed. In this running example, the path to achieve the goal of the CMS actor is by means of the navigational requirement *"Blind review process"*, all the requirements implemented for this path are labeled with (✓). We can observe in the SR model that some navigational and service requirements are decomposed in other requirements, some of them affects positively or negatively some non-functional requirements, i.e., the service requirement *"Download paper without authors' name"* needs the content requirement *"Papers"*, also, affects positively the softgoal *"Privacy be maximized"* and in some negatively form the softgoal *"Obtain more complete info"*. This fact is very important to see how to satisfy the goal *"Process of review of papers be selected"* considering the Web application softgoals. Therefore, maximizing or minimizing the contribution from requirements to softgoals is a viable solution to find a path to fully-satisfy the goal.

4 An Impact Analysis Algorithm for Goal-Oriented Requirements in Web Engineering

In this section, we present a proposal which provides a way to analyze the impact of a change in an A-OOH conceptual model for the goal-oriented requirement approach for Web engineering described in the previous section. Therefore, the Web developer will be able to evaluate the effect of the change and select the best among several design options to fully satisfy goals based on maximizing softgoals.

The algorithm is designed to be applied in $i*$ requirements model considering the type of contributions made by the intentional elements to the softgoals. This algorithm allows to evaluate the impact in the requirements model resulting from removing any element of the A-OOH conceptual models. In this way, it is determined which new requirements should be included in the A-OOH conceptual models for maximizing softgoal satisfaction although some requirements have been removed. To this aim, some heuristics have been defined.

4.1 Heuristics

Some heuristics have been defined for determining the impact of the contribution links between intentional element and softgoals. Table 1 summarizes some terms for understanding the heuristics presented in this section. These terms correspond to the most common types of contribution links of the $i*$ modeling framework.

The *"Help"* contribution link is a partial positive contribution, not sufficient by itself to satisfy the softgoal. The contribution link named *"Hurt"* is partial negative contribution, not sufficient by itself to deny the softgoal. *"Some +"* is a positive contribution whose strength is unknown. *"Some -"* is the opposite contribution type to *"Some +"*, is a negative contribution whose strength is unknown. The *"Break"* contribution link refers to a negative contribution enough to deny a softgoal. Finally, the *"Make"* contribution link is a positive contribution strong enough to satisfy a softgoal.

Table 1. The *i** contributions types

Heuristics Terms	***i**** **Contribution Type**
Strongly-positive	Help
Weakly-positive	Some +
Strongly-negative	Hurt
Weakly-negative	Some -
Dependent-negative	Break
Dependent-positive	Make

The heuristics defined are:

– **H1.** If the contribution of the requirement to remove is *strongly-positive*, and the contribution of the requirement to implement is *strongly-negative*, **do not** implement the requirement.
– **H2.** If the contribution of the requirement to remove is more *strongly-positive* than the contribution of the requirement to implement, but the contribution to be implemented is *weakly-negative*, the requirement **could be** implemented.
– **H3.** If the contribution of the requirement to remove is *strongly-negative* or *weakly-negative*, and the contribution of the requirement to implement is *strongly-positive* or *weakly-positive*, the requirement **should be** implemented.
– **H4.** If the polarity of the contribution of the requirement to remove is as negative as the contribution of the requirement to implement, the requirement to implement **should not be** implemented to maximize the satisfaction of the softgoal.
– **H5.** If the polarity of the contribution of the requirement to remove is as positive as the contribution of the requirement to implement, the requirement to implement **should be** implemented to maximize the satisfaction of the softgoal.
– **H6.** If the contribution of the requirement to remove is *Dependent-positive*, then the developer should consider whether that requirement should be removed or not considering the need to implement this softgoal.
– **H7.** If the contribution of the requirement to remove is *Dependent-negative*, then the developer should consider whether that requirement should be removed or not considering the impact of not implementing this softgoal.

4.2 Preconditions and Postconditions

The preconditions should be launched before the execution of the algorithm and can be only applied to the elements implemented in the requirements model described in Section 3. Specifically, these preconditions permit the execution of the algorithm when:

1. The requirement to remove does not affect the goal by the *"means-end"* contribution type.
2. When there is more than one *"means-end"* contribution type, it means that the impact analysis will be possible by means of the softgoals tradeoff.
3. The requirement to remove affects other requirements and these requirements (not shared) are not in the possible paths to satisfy the goal.

Moreover, there is one postcondition to be applied to the elements defined in the i^* requirements model. This postcondition is applied when a requirement has been selected to be implemented in the requirements model as an alternative solution for the satisfaction of the goal: if the requirement to be implemented has associated requirements, these requirements must be implemented automatically.

4.3 Impact Analysis Algorithm

This algorithm considers the contributions made by the intentional elements from the requirements model to find a path to fully satisfy, where possible, the main goal. To do this, the designer must have to find tradeoffs between the softgoals. The algorithm is presented next.

```
FUNCTION TradeOffAlgorithm (RequirementsModel)
    TR= task to remove; TI= task to implement;
    SN= new softgoal; IEList= intentional elements list;
    ASList= affected softgoals list;
    TIList= list of task to implement; Value= false;
    P = PreConditions();
    IF (P=true) THEN
      IEList= CreateIntentionalElementsList(RequirementsModel);
      IF(TR.Contributes2Softgoals()) THEN
         ASList=CreateAffectedSoftgoalsList(TR);
         FOREACH s FROM ASList:
             TI= SearchTaskToApply(IEList);
             Value= Heuristics(ASList,IEList,TI);
             TI.AddValue(Value);
             TIList.Add(TI);
                IF (TI.Contributes2Softgoals(TI)) THEN
                  ASList.add(SN);
                END IF
         END FOREACH
         FOREACH v FROM TIList:
             CalculateAverage(v);
             IF(CalculateAverage(v)) THEN
```

```
23                 Implements(v);
24              END IF
25           END FOREACH
26        END IF
27     PostCondition();
28     ELSE
29        ShowMessage(P.message());
30     END IF
31  END PROGRAM
```

Algorithm 1.1. Algorithm for impact analysis in goal-oriented Web engineering

A snippet of code that represents our algorithm for impact analysis of goal-oriented requirements in Web engineering is shown in 1.1. First, in lines 6 and 7 the pre-conditions are evaluated (Section 4.2), these must be "true" to proceed with the execution of the algorithm. Next, from lines 8 to 19, the algorithm creates a list of intentional elements. In these lines, all types of requirements from the requirements model are stored in this list. The next step is to extract those softgoals that receive a contribution from the requirement to remove, for each softgoal from the list, finding a non-implemented requirement and applying the heuristics introduced in Section 4.1. Each of these requirements must be stored in the list, and if it contributes to a softgoal, the softgoal must be stored in the list too. Then, lines 20 to 29 are used to evaluate each element from the list according to the weight of each element assigned by the heuristics to determine when a requirement must be implemented. Finally, the postcondition is executed and the alternative path to fully satisfy the goal from requirements model is obtained.

5 Performing the Impact Analysis

Lets suppose the following scenario: the Web developer decides deleting from the domain model of A-OOH the elements that correspond to the requirement *"Download papers without authors' name"*. It is necessary to know which other requirements are affected by this change. In addition, this action implies that the goal *"Process of review of papers be selected"* can not be satisfied. Thus, it is necessary to search for alternative paths in the $i*$ requirements model (if there any) in order to fully-satisfy the goal *"Process of review papers be selected"*. To this aim, our algorithm is triggered. The execution of the impact analysis algorithm is detailed next.

The first step to execute our algorithm consists of applying the preconditions. For this running example, the preconditions result true, it means that there is any problem to the algorithm has been executed.

Next, it is necessary to develop a list of the requirements (implemented or not) that contribute to any softgoal in the $i*$ requirements model (see Table 2). Also, if a softgoal contributes to other one, the softgoal must be added to the list too.

Table 2. The requirements contributions to softgoals

Requirements	*"S1"*	*"S2"*	*"S3"*	*"S4"*	*"S5"*	*"S6"*
"Blind review process"	Help	Break	Hurt	Help	-	-
"Download papers without authors' name"	**-**	**-**	**-**	**-**	**Help**	**Some -**
"Normal review process"	Some -	Make	Help	-	-	-
"Download paper with authors' name"	-	-	-	Hurt	Some -	Help
"View review process status"	-	-	-	-	-	Help
"Obtain more complete info"	-	-	Help	-	-	-

Table 2 highlights in bold the requirement to be removed. This table shows a requirements list (functional and non-functional) and their type of contributions to the softgoals where S1 corresponds to softgoal *"Be fair in review"* from requirements model, S2 to *"Review process easier"*, S3 represents *"Accurate review process"*, S4 conforms to *"Avoid possible conflicts of interest"*, S5 its the *"Privacy be maximized"* softgoal and S6 refers to *"Obtain more complete info"*.

The next step is to identify the number of softgoals affected by the requirement to be removed. If necessary, a list of the softgoals that receive a contribution from the requirement to be removed is made. In this example, the requirement to be removed is *"Download papers without authors' name"*, this one affects two softgoals: *"Privacy be maximized"* and *"Obtain more complete info"* S5 y S6 respectively (see Table 2).

For each softgoal that receives a contribution from the requirement to be removed, we search for a non-implemented requirement of which contribution compensates the possible elimination of the requirement to be removed. To do this, it is necessary to apply the heuristics defined in Section 4.1.

For example, the softgoal *"Privacy be maximized"*, according to Table 1, receives a *strongly-positive* contribution (Help) from the requirement to be removed, thus being necessary searching for a non-implemented requirement to contribute to this softgoal. In this case, only the requirement *"Download papers with authors' name"* contributes (negatively) to this softgoal (*weakly-negative*, i.e. Some -). Therefore, applying the heuristics described in Section 4.1, specifically the heuristic number 2 (H2), the requirement *"Download papers with authors' name"* could be implemented.

Considering the softgoal *"Obtain more complete info"* according to Table 1, it receives a *weakly-negative* contribution (Some -) from the requirement to be removed, thus being necessary searching for a non-implemented requirement to contribute to this softgoal. In this case, two requirements (positively) contribute to this softgoal, *"Download papers with authors' name"* and *"View review process status"* (*strongly-negative*, i.e. Help). Therefore, the heuristic H3 applies for this softgoal, thus, these requirements should be implemented.

After analyzing the softgoals contributions, the next step is searching for any softgoal in the requirements list that contributes to another softgoal. In this example, the softgoal *"Obtain more complete info"* makes a *strongly-positive* contribution (Help) to the softgoal *"Accurate review process"*, thus, the next step consists of searching for the requirement that makes a contribution to the softgoal and applying the heuristics. Therefore, the requirement that makes a contribution to the softgoal *"Accurate review process"* is *"Normal review process"*, this contribution is *strongly-positive* (see Figure 2), hence, according to H3 this requirement must be implemented.

After these steps, Table 3 shows requirements that could be implemented to fully-satisfy the goal *"Process of review papers be selected"* after having removed the requirement *"Download papers without authors' name"*. Next, it is necessary to evaluate the heuristics assigned to each requirement to know what could be implemented.

Table 3. Non-implemented requirements that contributes to softgoals

Intentional element	*"S5"*	*"S4"*	*"S6"*	*"S3"*	*Result*
"Download papers with authors' name"	H2 (Some -)	H1 (Hurt)	H3 (Help)	-	Implement
"Normal review process"	-	-	-	H3 (Help)	Implement
"View review process status"	-	-	H3 (Help)	-	Implement

Table 3 shows the results after having performed the algorithm. In this table the requirements that must be implemented in order to fully-satisfy the goal *"Process of review papers be selected"* are shown. To do this, it is necessary to evaluate the contribution type of each requirement, i.e., the navigational requirement *"Download papers with authors' name"* negatively contributes (Some -) to the softgoal *"Privacy be maximized"*, thus hurting the softgoal *"Avoid possible conflicts of interest"* and helping the softgoal *"Obtain more complete info"*. Therefore, by using the human judgment the navigational requirement *"Download papers with authors' name"* can be implemented. For the navigational requirement *"Normal review process"*, it is easier to determine whether it can be implemented because it only contributes to one softgoal, the *"Accurate review process"*, hence its contribution is Help, therefore this requirement must be implemented. Finally, the navigational requirement *"View review process status"* positively contributes to the softgoal *"Obtain more complete info"*, consequently this requirement must be implemented.

The final step is to apply the postcondition from Section (4.2). In this running example, according to the postcondition, it is necessary to implement the navigational requirements *"View papers info"* and *"View Accepted/Rejected papers"* because these requirements are associated with the navigational requirement *"View Review Process Status"*. In addition, the content requirement *"Authors"* and the service requirement *"Send Comments to Authors"* must be implemented too in order to implement the alternative path to fully satisfy the goal *"Process of review papers be selected"*. Hence, the content requirement *"Authors"* is associated with the navigational requirement *"View Accepted/Rejected papers"* and the service requirement *"Send Comments to Authors"* is related with the navigational requirement *"Normal review process"*.

After finishing the execution of the algorithm, we obtain the requirements that are directly and indirectly affected by the deletion of the requirement *"Download papers without authors' name"*. Moreover, the algorithm can find out which requirements must be implemented to continue satisfying the goal considering the contributions received from the softgoals. In this running example the

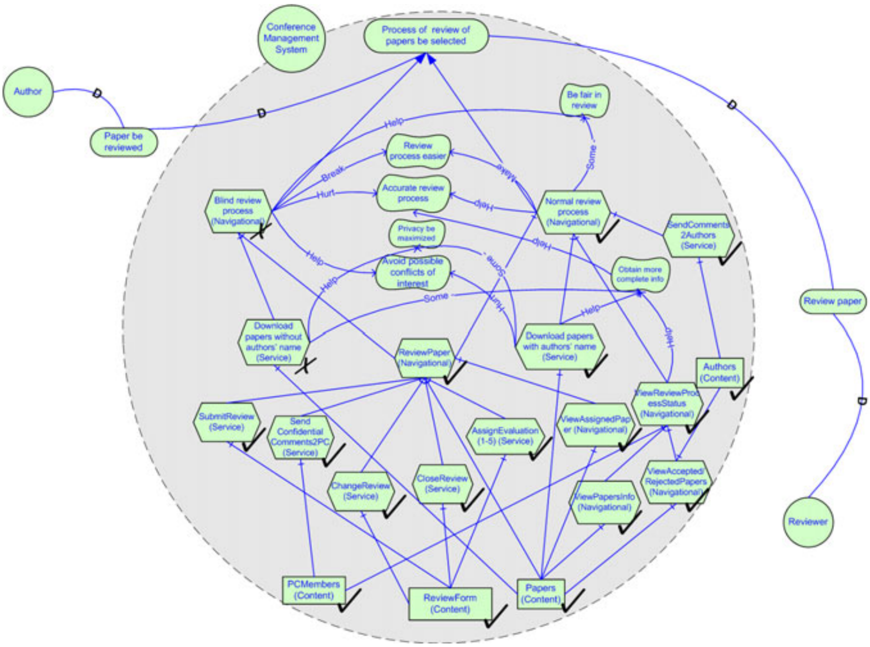


Fig. 3. Conference Management System requirements expressed in a SR and SD Models with the alternative path to fully satisfy the goal *"Process of review papers be selected"*

requirements to implement are: *"Download papers with authors' name"*, *"Normal review process"* and *"View review process status"*. Finally, according to the post-condition the requirements *"View papers info"*, *"View Accepted/Rejected papers"*, *"Authors"* and *"Send Comments to Authors"* must be implemented too. Figure 3 shows the final requirements model with the alternative path implemented to fully-satisfy the goal *"Process of review papers be selected"*.

6 Conclusions and Future Work

Due to the dynamic idiosyncrasy of the Web and its heterogeneous audience Web applications should consider a requirement analysis phase in order to reflect, from the early stages of the Web application development process, specific needs, goals, interests and preferences of each user or user type, and due to the fast evolution of the Web, possible changes in those requirements should also be managed.

In this work, we have presented a methodology based on the i^* modelling framework to specify Web requirements. Moreover, an algorithm to analyze the impact derived from a change done in the requirements model is presented.

Benefits of applying the algorithm described in this work include both the analysis of the impact derived from a change in a conceptual model and the ability to find an alternative path to fully-satisfy the goal by means of the softgoals tradeoff.

Nevertheless, according to [7], the softgoals are not considered with sufficient importance from the early stages of the development process. In this context, our proposal makes a contribution to the requirements analysis field considering the softgoals, hence it allows the designer to make decisions from the very beginning stages of the development process that would affect the structure of the envision website in order to satisfy users needs. Therefore, the designer can improve the quality of the requirements model analyzing the balance of the softgoals with the stakeholders.

Our short-term future work includes the definition of a metamodel to help to record the relationships among functional and non-functional requirements to adapt the tradeoff algorithm presented in this work.

Finally, note that this work has been done in the context of the A-OOH modeling method, however it can be applied to any Web modeling approach.

Acknowledgments. This work has been partially supported by the following projects: MANTRA (GV/2011/035) from Valencia Ministry, MANTRA (GRE09-17) from the University of Alicante, SERENIDAD (PEII-11-0327-7035) from Junta de Comunidades de Castilla La Mancha (Spain) and by the MESOLAP (TIN2010-14860) project from the Spanish Ministry of Education and Science. José Alfonso Aguilar is subventioned by CONACYT (Consejo Nacional de Ciencia y Tecnología) Mexico and University of Sinaloa, Mexico.

References

1. Ginige, A.: Web engineering: managing the complexity of web systems development. In: SEKE, pp. 721–729 (2002)
2. Zhang, W., Mei, H., Zhao, H.: A feature-oriented approach to modeling requirements dependencies (2005)
3. Arnold, R., Bohner, S.: Impact analysis-towards a framework for comparison. In: Proceedings of Conference on Software Maintenance, CSM 1993, pp. 292–301. IEEE, Los Alamitos (2002)
4. Lindvall, M., Sandahl, K.: How well do experienced software developers predict software change? Journal of Systems and Software 43(1), 19–27 (1998)
5. Zhang, S., Gu, Z., Lin, Y., Zhao, J.: Celadon: A change impact analysis tool for Aspect-Oriented programs. In: Companion of the 30th International Conference on Software Engineering, pp. 913–914. ACM, New York (2008)
6. Gupta, C., Singh, Y., Chauhan, D.: Dependency based Process Model for Impact Analysis: A Requirement Engineering Perspective. International Journal 6
7. Ameller, D., Gutiérrez, F., Cabot, J.: Dealing with non-functional requirements in model-driven development. In: 18th IEEE International Requirements Engineering Conference, RE (2010)
8. Sommerville, I.: Software Engineering, 6th edn. Addison-Wesley, Reading (2001)
9. Boehm, B., In, H.: Identifiing Quality-Requirement Conflicts (2002)

10. Elahi, G., Yu, E.: Modeling and analysis of security trade-offs - a goal oriented approach. Data and Knowledge Engineering 68(7), 579–598 (2009); Special Issue: 26th International Conference on Conceptual Modeling (ER 2007) - Six selected and extended papers
11. Nuseibeh, B., Easterbrook, S.M.: Requirements engineering: a roadmap. In: ICSE - Future of SE Track, pp. 35–46 (2000)
12. Bolchini, D., Mylopoulos, J.: From task-oriented to goal-oriented web requirements analysis. In: WISE 2003: Proceedings of the Fourth International Conference on Web Information Systems Engineering, p. 166. IEEE Computer Society, Washington, DC, USA (2003)
13. Aguilar, J.A., Garrigós, I., Mazón, J.N., Trujillo, J.: An mda approach for goal-oriented requirement analysis in web engineering. J. UCS 16(17), 2475–2494 (2010)
14. Garrigós, I., Mazón, J.-N., Trujillo, J.: A requirement analysis approach for using i* in web engineering. In: ICWE, pp. 151–165 (2009)
15. Aguilar, J.A., Garrigós, I., Mazón, J.N., Trujillo, J.: Web Engineering approaches for requirement analysis- A Systematic Literature Review. In: 6th Web Information Systems and Technologies (WEBIST), Valencia, Spain, vol. 2, pp. 187–190. SciTePress Digital Library (2010)
16. Schwabe, D., Rossi, G.: The object-oriented hypermedia design model. Communications of the ACM 38(8), 45–46 (1995)
17. De Troyer, O.M.F., Leune, C.J.: Wsdm: a user centered design method for web sites. Comput. Netw. ISDN Syst. 30(1-7), 85–94 (1998)
18. Casteleyn, S., Van Woensel, W., Houben, G.-J.: A semantics-based aspect-oriented approach to adaptation in web engineering. In: Hypertext, pp. 189–198 (2007)
19. Fons, J., Valderas, P., Ruiz, M., Rojas, G., Pastor, O.: Oows: A method to develop web applications from web-oriented conceptual models. In: International Workshop on Web Oriented Software Technology (IWWOST), pp. 65–70 (2003)
20. Ceri, S., Fraternali, P., Bongio, A.: Web modeling language (webml): a modeling language for designing web sites. The International Journal of Computer and Telecommunications Networking 33(1-6), 137–157 (2000)
21. Escalona, M.J., Aragón, G.: Ndt. a model-driven approach for web requirements. IEEE Transactions on Software Engineering 34(3), 377–390 (2008)
22. Koch, N.: The expressive power of uml-based web engineering. In: International Workshop on Web-oriented Software Technology (IWWOST), pp. 40–41 (2002)
23. Bolchini, D., Paolini, P.: Goal-driven requirements analysis for hypermedia-intensive web applications, vol. 9, pp. 85–103. Springer, Heidelberg (2004)
24. Molina, F., Pardillo, J., Toval, A.: Modelling web-based systems requirements using wrm. In: Web Information Systems Engineering (WISE) Workshops, pp. 122–131. Springer, Heidelberg (2008)
25. Yu, E.: Modelling Strategic Relationships for Process Reenginering. PhD thesis, University of Toronto, Canada (1995)
26. Yu, E.: Towards modeling and reasoning support for early-phase requirements engineering. In: RE, pp. 226–235 (1997)
27. Escalona, M.J., Koch, N.: Requirements engineering for web applications - a comparative study. J. Web Eng. 2(3), 193–212 (2004)
28. Molina, F., Toval, A.: Integrating usability requirements that can be evaluated in design time into model driven engineering of web information systems. Adv. Eng. Softw. 40, 1306–1317 (2009)
29. Horkoff, J., Yu, E.: Evaluating Goal Achievement in Enterprise Modeling–An Interactive Procedure and Experiences. The Practice of Enterprise Modeling, 145–160 (2009)

30. Horkoff, J., Yu, E.: A Qualitative, Interactive Evaluation Procedure for Goal-and Agent-Oriented Models. In: CAiSE Forum
31. Aguilar, J.A., Garrigós, I., Mazón, J.-N.: Modelos de weaving para trazabilidad de requisitos web en a-ooh. In: DSDM: Actas del VII Taller sobre Desarrollo de Software Dirigido por Modelos, JISBD, Congreso Espanol de Informatica (CEDI), Valencia, Espana, SISTEDES, pp. 146–155. SISTEDES (2010)
32. Garrigós, I.: A-OOH: Extending Web Application Design with Dynamic Personalization. PhD thesis, University of Alicante, Spain (2008)
33. Eclipse (2010), http://www.eclipse.org/
34. Pastor, O.: Conference procedings. In: IWWOST (2001)

An Improved Protocol for Server-Aided Authenticated Group Key Establishment*

Junghyun Nam[1], Juryon Paik[2], Byunghee Lee[2], Kwangwoo Lee[2], and Dongho Won[2,**]

[1] Department of Computer Engineering, Konkuk University, Korea
jhnam@kku.ac.kr

[2] Department of Computer Engineering, Sungkyunkwan University, Korea
wise96@ece.skku.ac.kr, {bhlee,kwlee,dhwon}@security.re.kr

Abstract. Protocols for group key establishment enable a group of parties to build a secure multicast channel over insecure public networks. In this paper, we present a group key transfer protocol designed for use in the model where a trusted key generation center shares a long-term secret with each of its registered users. Our protocol is an improved version of the group key transfer protocol proposed recently by Harn and Lin. Improvement is made in three different aspects: security, efficiency and correctness. Our main contribution is in showing that Harn and Lin's protocol does not achieve implicit key authentication but can be fixed without causing any efficiency degradation.

Keywords: Security, key establishment protocol, multicast, group key transfer, secret sharing.

1 Introduction

Key establishment protocols are distributed algorithms that describe how two or more communicating parties generate a common secret key called a *session key*. Session-key generation is one of the fundamental cryptographic operations and provides a typical way of building secure communication channels over insecure public networks. Traditionally, protocols which can be run by an arbitrary number of parties are called group key establishment protocols, in contrast to protocols which can be run only by two or three parties. In the group setting, a session key is also called a *group key*. Key establishment protocols are often classified into two types [3]: *key agreement protocols* and *key transfer protocols*. Key agreement protocols require each participant to contribute its part to the final form of the session key, whereas key transfer protocols allow one trusted entity to generate the session key and then transfer it to all participants.

* This work was supported by Priority Research Centers Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0020210).

** Corresponding author.

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 437–446, 2011.

Over the past several years, there has been a tremendous surge of interest in group-oriented applications where a group of parties communicate collaboratively to achieve their common interest or objective. Typical group-oriented applications include distributed multiplayer games, grid computing, collaborative workspaces, video/audio teleconferencing, and social networking services. In particular, social networking services such as Twitter [24] and Facebook [9] have recently gained widespread popularity and are redefining our sense of community. The proliferation of group-oriented applications has led to a growing concern in security of group communications. The current Internet, by design, is an open network which might be controlled by an adversary. Today's adversaries are equipped with more powerful computing resources and attacking tools than ever before. The situation gets even worse when we consider malicious insiders. In general, we cannot expect complete trust among all group members just because they collaborate to achieve a specific purpose; collaboration does not imply full trust. Perhaps malicious insiders pose the most serious security threat to many organizations and enterprises.

Protocols for group key establishment are valuable tools for protecting group communications. A set of parties communicating over a public network can generate a common session key by running a group key establishment protocol. Once a session key has been established, the parties can use this key to encrypt and/or authenticate their subsequent multicast messages. This represents a typical way of achieving secure group communications. The session key, of course, must be known only to the intended parties at the end of the protocol run, because otherwise the whole system becomes vulnerable to all manner of attacks. Roughly stated, a key exchange protocol satisfying this requirement is said to be *authenticated*. Due to their significance in building secure multicast channels, authenticated group key establishment protocols have been extensively studied over the last decades [11,7,17,15,6,16,5,13,4,14,12,2,18,1,25]. However, despite all the research efforts made so far, the design of an authenticated group key establishment protocol is still notoriously hard. Many group key establishment protocols, even some with a claimed proof of security, have been analyzed to be vulnerable to a certain kind of attack years after they were published [21,8,23,19,20]. Hence, group key establishment protocols must be subjected to a thorough scrutiny before they can be deployed into a public network.

Recently, Harn and Lin [10] proposed a group key transfer protocol based on Shamir's secret sharing [22]. Harn and Lin's protocol assumes a trusted key generation center (KGC) who provides key distribution service to its registered users. In this paper, we propose a group key transfer protocol which improves Harn and Lin's protocol. Our improvement is made in three different aspects: security, efficiency and correctness. Firstly, we show that Harn and Lin's protocol fails to achieve implicit key authentication. Fortunately, this security weakness can be effectively addressed without causing any efficiency degradation. Secondly, we show that the scalability limitation of Harn and Lin's protocol can be somewhat mitigated by reducing the length of the broadcast message sent by KGC. Finally, we show incorrectness of Harn and Lin's protocol and present how to make the protocol correct.

2 Review of the HL Protocol

In this section, we review the group key transfer protocol proposed recently by Harn and Lin [10]. We use HL to refer to Harn and Lin's protocol. The protocol HL employs Shamir's secret sharing [22] to achieve information-theoretically secure distribution of session keys. The protocol HL consists of three phases: system initialization, user registration, and key distribution.

System initialization. KGC randomly chooses two safe primes p and q (i.e., p and q are primes such that $p' = \frac{p-1}{2}$ and $q' = \frac{q-1}{2}$ are also primes) and computes $n = pq$. n is made publicly known.

User registration. Each user is required to register at KGC to subscribe the key distribution service. During registration, KGC shares a secret (x_i, y_i) with each user U_i where $x_i, y_i \in \mathbb{Z}_n^*$.

Key distribution. This phase constitutes the core of the protocol and is performed whenever a group of users $U_1, \ldots, U_t$ decide to establish a common session key.

Step 1. A designated user of the group, called the initiator, sends a key distribution request to KGC. The request carries the list of participating users $\langle U_1, \ldots, U_t \rangle$.

Step 2. KGC broadcasts the participant list $\langle U_1, \ldots, U_t \rangle$ in response to the request.

Step 3. Each user U_i, for $i = 1, \ldots, t$, sends a random challenge $r_i \in \mathbb{Z}_n^*$ to KGC.

Step 4. KGC randomly selects a session key k and constructs by interpolation a t-th degree polynomial $f(x)$ passing through the $(t+1)$ points: $(x_1, y_1 \oplus r_1), \ldots, (x_t, y_t \oplus r_t)$ and $(0, k)$. Next, KGC selects t additional points $P_1, \ldots, P_t$ that lie on the polynomial $f(x)$. KGC then computes $\beta = h(k, U_1, \ldots, U_t, r_1, \ldots, r_t, P_1, \ldots, P_t)$, where h is a one-way hash function, and broadcasts β and $\langle r_1, \ldots, r_t, P_1, \ldots, P_t \rangle$ to the users. All computations with respect to $f(x)$ are performed modulo n.

Step 5. Each U_i constructs the polynomial $f(x)$ from the $(t+1)$ points: $P_1, \ldots, P_t$ and $(x_i, y_i \oplus r_i)$. Then U_i recovers the session key $k = f(0)$ and checks the correctness of β in the straightforward way. U_i aborts if the check fails.

The main security goals that HL intends to achieve are: (1) key freshness, (2) key confidentiality, and (3) key authentication. Key freshness is to ensure that a session key has never been used before. Key confidentiality means ensuring that the session key can be computed by authorized users only. Key authentication provides assurance to authorized users that the session key is distributed by KGC, but not by an adversary.

The HL protocol

KGC: key generation center
$U_i \in \{U_1, \ldots, U_t\}$
h : one-way hash function
$\dashrightarrow$: broadcast

KGC		U_i
$\langle (x_1, y_1) \ldots, (x_t, y_t) \rangle$		(x_i, y_i)
	$\xleftarrow{\langle U_1, \ldots, U_t \rangle}$	
	$\xrightarrow{\langle U_1, \ldots, U_t \rangle}$ (broadcast)	
		$r_i \in \mathbb{Z}_n^*$
		$\alpha_i = h(x_i, y_i, r_i)$
	$\xleftarrow{\langle \alpha_i, r_i \rangle}$	
check $\alpha_1, \ldots, \alpha_t$		
determine the session key k		
construct $f(x)$		
select $P_1, \ldots, P_t$		
$\beta = h(k, U_1, \ldots, U_t, r_1, \ldots, r_t, P_1, \ldots, P_t)$		
	$\xrightarrow{\beta, \langle r_1, \ldots, r_t, P_1, \ldots, P_t \rangle}$ (broadcast)	
		construct $f(x)$
		recover $k = f(0)$
		check β

Fig. 1. The HL protocol

The protocol HL does not take into account authentication of user messages but focuses only on achieving the three security goals stated above. Notice that user messages, service requests and random challenges, are not authenticated at all. This means that an adversary may impersonate a legitimate user to request for a key-distribution service. To avoid this situation, the authors of [10] suggested (in Remark 2) a way of incorporating user/message authentication into the protocol. Their suggestion is to revise Steps 3 and 4 as follows:

Step 3 (revision). Each user U_i, for $i = 1, \ldots, t$, selects a random challenge $r_i \in \mathbb{Z}_n^*$, computes $\alpha_i = h(x_i, y_i, r_i)$, and sends $\langle \alpha_i, r_i \rangle$ to KGC.

Step 4 (revision). KGC checks the correctness of each α_i in the straightforward way. KGC aborts if any of the checks fails. Otherwise, KGC continues with the original version of Step 4.

Fig. 1 illustrates (the key distribution phase of) the protocol HL.

3 Analysis of HL

This section analyzes and improves the protocol HL in three aspects: security, efficiency and correctness.

3.1 Security

Implicit key authentication is the fundamental security property that any given key establishment protocol is expected to achieve. Informally, this property means that no one outside the group can gain access to the session key. More formally:

Definition 1 (implicit key authentication). *Let $\mathcal{G}$ be a set of users who wish to share a session key by running a key establishment protocol P. Let sk_i be the session key computed by a user $U_i \in \mathcal{G}$ as a result of an execution of protocol P. We say that P achieves implicit key authentication if each $U_i \in \mathcal{G}$ is assured that no $U_k \notin \mathcal{G}$ can learn the key sk_i unless helped by a dishonest $U_j \in \mathcal{G}$.*

A key exchange protocol achieving implicit key authentication is said to be *authenticated*, and is a primitive of crucial importance in much of modern cryptography and network security.

Our main result is that the protocol HL fails to achieve authenticated key establishment. We prove this result by giving an active attack that violates implicit key authentication of HL. The adversary U_A who mounts the attack is a legitimate user in the sense that she is registered with KGC and thus is able to set up normal protocol sessions with other users. The attack proceeds as follows:

1. When the initiator sends a key distribution request along with the participant list $\langle U_1, \ldots, U_t \rangle$, the adversary U_A intercepts the request message and instead sends its own key distribution request alleging that the participants are $U_A, U_2, \ldots, U_t$.

2. KGC will broadcast the participant list $\langle U_A, U_2, \ldots, U_t\rangle$ in response to U_A's request. But, U_A replaces $\langle U_A, U_2, \ldots, U_t\rangle$ with $\langle U_1, \ldots, U_t\rangle$.
3. U_A intercepts the message $\langle \alpha_1, r_1\rangle$ sent by U_1 to KGC. (But, the messages sent by $U_2, \ldots, U_t$ are transmitted without any interruption.) Then, U_A selects a random challenge $r_A \in \mathbb{Z}_n^*$, computes $\alpha_A = h(x_A, y_A, r_A)$, and sends $\langle \alpha_A, r_A\rangle$ to KGC.
4. KGC cannot detect any discrepancy since α_A is valid. Hence, KGC will broadcast the keying material $\langle \beta, r_1, \ldots, r_t, P_1, \ldots, P_t\rangle$ to the users, where $\beta = h(k, U_A, U_2, \ldots, U_t, r_A, r_2, \ldots, r_t, P_1, \ldots, P_t)$. U_A intercepts this broadcast message and proceeds to forge the keying material as follows: U_A first constructs the polynomial $f(x)$ and recovers the session key $k = f(0)$ as specified in the protocol. U_A then computes $\beta' = h(k, U_1, \ldots, U_t, r_1, \ldots, r_t, P_1, \ldots, P_t)$ and broadcasts $\langle \beta', r_1, \ldots, r_t, P_1, \ldots, P_t\rangle$ as if it is from KGC.
5. After receiving the broadcast message from U_A, the users $U_2, \ldots, U_t$ will construct the polynomial $f(x)$ and recover the session key $k = f(0)$ which is already known to U_A.

At the end of the attack, each of $U_2, \ldots, U_t$ believes that they have established a secret session key only with the rest of the group, while in fact the key has been shared also with the adversary U_A. This demonstrates that the protocol HL does not provide implicit key authentication and therefore fails to achieve authenticated key establishment.

Fortunately, the attack above can be effectively prevented. As a simple countermeasure, we recommend to modify Step 3 as follows:

Step 3. Each user U_i, for $i = 1, \ldots, t$, selects a random challenge $r_i \in \mathbb{Z}_n^*$, computes $\alpha_i = h(x_i, y_i, r_i, U_1, \ldots, U_t)$, and sends $\langle U_i, \alpha_i, r_i\rangle$ to KGC.

The identities of all protocol participants are now included as part of the messages to be hashed. With this modification applied, the messages sent by $U_2, \ldots, U_t$ in Step 3 of the attack cannot pass the verification test by KGC. Hence, the attack is no longer valid against the improved protocol.

3.2 Efficiency

The major shortcoming of HL is that the length of the broadcast message (sent in Step 4) increases significantly as the group size grows. Harn and Lin recognized this shortcoming and thus stated that their protocol is only suitable for groups of small size. But, we found that this scalability limitation of HL can be somewhat mitigated by a simple tweak. Let r_0 be a random challenge selected by KGC. Then we suggest that KGC should construct the polynomial $f(x)$ from $(x_1, y_1 r_0), \ldots, (x_t, y_t r_0)$ and $(0, k)$ instead of from $(x_1, y_1 \oplus r_1), \ldots, (x_t, y_t \oplus r_t)$ and $(0, k)$. This way, KGC need neither receive nor broadcast the random challenges $r_1, \ldots, r_t$. Without the challenges $r_1, \ldots, r_t$, the length of the broadcast message can be reduced up to about 30 percent. Notice that we have replaced the bitwise XOR operations by the multiplications. This replacement has been made due to that the multiplicative group $\mathbb{Z}_n^*$ is not closed under bitwise XOR operations (see the next subsection for the discussion on correctness of HL).

3.3 Correctness

Finally, we note that polynomial interpolation over $\mathbb{Z}_n^*$ may fail, though the probability of failure is negligible. This is essentially because the multiplicative group $\mathbb{Z}_n^*$ is not closed under addition and subtraction while interpolation formulas include additive and subtractive terms. If an addition/subtraction operation in $\mathbb{Z}_n^*$ returns a value z such that $\gcd(z, n) \neq 1$ (i.e., $z = cp$ or $z = cq$ for some integer c), then there will not exist the multiplicative inverse of z modulo n. The protocol HL fails if such a z happens to be a divisor in the interpolation formula. (Of course, the probability of this occurring should be negligible, because otherwise we have a polynomial-time factoring algorithm.) This correctness issue of HL can be addressed simply by replacing $\mathbb{Z}_n^*$ with a prime field $\mathbb{F}_p$ in which interpolation never fails. We believe that the replacement causes no security degradation.

4 The Proposed Protocol

Reflecting all the suggestions made in the previous section, we here propose an improved group key transfer protocol based on Shamir's secret sharing. Like HL, our improved protocol also consists of three phases: system initialization, user registration, and key distribution.

System initialization. KGC randomly chooses a prime p and makes it public.

User registration. Each user is required to register at KGC to subscribe the key distribution service. During registration, KGC shares a secret (x_i, y_i) with each user U_i where $x_i, y_i \in \mathbb{Z}_p^*$.

Key distribution. Let $U_1, \ldots, U_t$ be a group of users who wish to establish a common session key. Then, the users can establish their session key as follows:

Step 1. A designated user of the group, called the initiator, sends a key distribution request to KGC. The request carries the list of participating users $\langle U_1, \ldots, U_t \rangle$.

Step 2. KGC selects a random $r_0 \in \mathbb{Z}_p^*$ and broadcasts r_0 and $\langle U_1, \ldots, U_t \rangle$ in response to the request.

Step 3. Each user U_i, for $i = 1, \ldots, t$, computes $\alpha_i = h(x_i, y_i, r_0, U_1, \ldots, U_t)$ and sends $\langle U_i, \alpha_i \rangle$ to KGC.

Step 4. KGC checks the correctness of each α_i in the straightforward way. KGC aborts if any of the checks fails. Otherwise, KGC randomly selects a session key k and constructs by interpolation a t-th degree polynomial $f(x)$ passing through the $(t + 1)$ points: $(x_1, y_1 r_0), (x_2, y_2 r_0), \ldots, (x_t, y_t r_0)$ and $(0, k)$. Next, KGC selects t additional points $P_1, \ldots, P_t$ that lie on the polynomial $f(x)$. KGC then computes $\beta = h(k, U_1, \ldots, U_t, r_0, P_1, \ldots, P_t)$, where h is a one-way hash function, and broadcasts β and $\langle P_1, \ldots, P_t \rangle$ to the users. All computations with respect to $f(x)$ are performed modulo p.

Step 5. Each U_i constructs the polynomial $f(x)$ from the $(t+1)$ points: $P_1, \ldots, P_t$ and $(x_i, y_i r_0)$. Then U_i recovers the session key $k = f(0)$ and checks the correctness of β in the straightforward way. U_i aborts if the check fails.

A high level description of our proposed protocol is provided in Fig. 2.

Our improved version of the HL protocol

KGC: key generation center
$U_i \in \{U_1, \dots, U_t\}$
h : one-way hash function
$\dashrightarrow$: broadcast

$\boxed{\text{KGC}}$ $\langle (x_1, y_1) \dots, (x_t, y_t) \rangle$

$\boxed{U_i}$ (x_i, y_i)

$\xleftarrow{\langle U_1, \dots, U_t \rangle}$

$r_0 \in \mathbb{Z}_p^*$

$\dashrightarrow$ $r_0, \langle U_1, \dots, U_t \rangle$

$\alpha_i = h(x_i, y_i, r_0, U_1, \dots, U_t)$

$\xleftarrow{\langle U_i, \alpha_i \rangle}$

check $\alpha_1, \dots, \alpha_t$
determine the session key k
construct $f(x)$
select $P_1, \dots, P_t$
$\beta = h(k, U_1, \dots, U_t, r_0, P_1, \dots, P_t)$

$\dashrightarrow$ $\beta, \langle P_1, \dots, P_t \rangle$

construct $f(x)$
recover $k = f(0)$
check β

Fig. 2. An improved version of the protocol HL

References

1. Abdalla, M., Bohli, J.-M., González Vasco, M.I., Steinwandt, R. (Password) authenticated key establishment: From 2-party to group. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 499–514. Springer, Heidelberg (2007)
2. Abdalla, M., Bresson, E., Chevassut, O., Pointcheval, D.: Password-based group key exchange in a constant number of rounds. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 427–442. Springer, Heidelberg (2006)
3. Boyd, C.: On key agreement and conference key agreement. In: Mu, Y., Pieprzyk, J.P., Varadharajan, V. (eds.) ACISP 1997. LNCS, vol. 1270, pp. 294–302. Springer, Heidelberg (1997)
4. Boyd, C., Nieto, J.: Round-optimal contributory conference key agreement. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 161–174. Springer, Heidelberg (2002)
5. Bresson, E., Chevassut, O., Pointcheval, D.: Group Diffie-Hellman key exchange secure against dictionary attacks. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 497–514. Springer, Heidelberg (2002)
6. Bresson, E., Chevassut, O., Pointcheval, D., Quisquater, J.-J.: Provably authenticated group Diffie-Hellman key exchange. In: 8th ACM Conference on Computer and Communications Security (CCS 2001), pp. 255–264 (2001)
7. Burmester, M., Desmedt, Y.: A secure and efficient conference key distribution system. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 275–286. Springer, Heidelberg (1995)
8. Choo, K.-K., Boyd, C., Hitchcock, Y.: Errors in computational complexity proofs for protocols. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 624–643. Springer, Heidelberg (2005)
9. Facebook, `http://www.facebook.com`
10. Harn, L., Lin, C.: Authenticated group key transfer protocol based on secret sharing. IEEE Transactions on Computers 59(6), 842–846 (2010)
11. Ingemarsson, I., Tang, D., Wong, C.: A conference key distribution system. IEEE Transactions on Information Theory 28(5), 714–720 (1982)
12. Katz, J., Shin, J.: Modeling insider attacks on group key-exchange protocols. In: 12th ACM Conference on Computer and Communications Security (CCS 2005), pp. 180–189 (2005)
13. Katz, J., Yung, M.: Scalable protocols for authenticated group key exchange. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 110–125. Springer, Heidelberg (2003)
14. Kim, H., Lee, S., Lee, D.: Constant-round authenticated group key exchange for dynamic groups. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 245–259. Springer, Heidelberg (2004)
15. Kim, Y., Perrig, A., Tsudik, G.: Simple and fault-tolerant key agreement for dynamic collaborative groups. In: ACM Conference on Computer and Communications Security (CCS 2000), pp. 235–244 (2000)
16. Kim, Y., Perrig, A., Tsudik, G.: Communication-efficient group key agreement. In: IFIP SEC 2001, pp. 229–244 (2001)
17. Mayer, M., Yung, M.: Secure protocol transformation via "Expansion": From two-party to groups. In: 6th ACM Conference on Computer and Communications Security (CCS 1999), pp. 83–92 (1999)

18. Nam, J., Paik, J., Kim, U., Won, D.: Constant-round authenticated group key exchange with logarithmic computation complexity. In: Katz, J., Yung, M. (eds.) ACNS 2007. LNCS, vol. 4521, pp. 158–176. Springer, Heidelberg (2007)
19. Nam, J., Paik, J., Kim, U., Won, D.: Security enhancement to a password-authenticated group key exchange protocol for mobile ad-hoc networks. IEEE Communications Letters 12(2), 127–129 (2008)
20. Nam, J., Paik, J., Won, D.: A security weakness in Abdalla et al.'s generic construction of a group key exchange protocol. Information Sciences 181(1), 234–238 (2011)
21. Pereira, O., Quisquater, J.-J.: A security analysis of the Cliques protocols suites. In: 14th IEEE Computer Security Foundations Workshop, pp. 73–81 (2001)
22. Shamir, A.: How to share a secret. Communications of the ACM 22(11), 612–613 (1979)
23. Shim, K., Woo, S.: Cryptanalysis of tripartite and multi-party authenticated key agreement protocols. Information Sciences 177(4), 1143–1151 (2007)
24. Twitter, http://twitter.com
25. Yi, X., Tso, R., Okamoto, E.: ID-Based group password-authenticated key exchange. In: Takagi, T., Mambo, M. (eds.) IWSEC 2009. LNCS, vol. 5824, pp. 192–211. Springer, Heidelberg (2009)

Presence Based Secure Instant Messaging Mechanism for IP Multimedia Subsystem

Zeeshan Shafi Khan[1], Muhammad Sher[2], and Khalid Rashid[2]

[1] Center of Excellence in Information Assurance, King Saud University, Saudi Arabia
[2] International Islamic University, Islamabad, Pakistan
zeeshanshafikhan@gmail.com

Abstract. Presence and instant messaging are the two important services offered on top of IP Multimedia subsystem. Instant messages can be session based or immediate. In both cases there are some security threats like session modification, session termination, invite flooding and message flooding, etc. In this paper, a presence based solution to mitigate the effect of these attacks is proposed. Results show that the proposed solution performs well to secure the instant messaging.

Keywords: Presence, Instant Messaging, IP Multimedia Subsystem.

1 Introduction

IP multimedia subsystem (IMS) provides architecture for fixed mobile convergence (FMC) that includes variety of protocols. Session initiation protocol (SIP), standardized by the International Engineering Task Force (IETF), is the main signaling protocol used by IMS. IMS can be divided into three layers. The lower layer is called as user plan and accommodates all IMS users who want to use different IMS services. To obtain services, an IMS user contacts IMS core where authentication and authorization takes place. IMS core mainly consists of call session control functions (CSCFs) and databases. Proxy call session control function (P-CSCF) acts as an entry point to IMS core and forwards the request to interrogating call session control function (I-CSCF). I-CSCF locates the appropriate serving call session control function (S-CSCF) from the databases and forwards the request to that S-CSCF. S-CSCF performs authentication by downloading the authentication data from the home subscriber server (HSS) and forwards the request to an application server if required. The service function includes the media resource function controller (MRFC), media resource function processor (MRFP), and application servers (AS) [1]. The upper layer of the IMS contains application servers to provide different kind of services.

Messaging service allows users to exchange messages with each other. Message contents can carry text or multimedia data. Immediate messaging and session based messaging are two main types of IMS messaging. In immediate messaging, the sender prepares a message, selects the destination, and forwards the message to that destination. If the receiver device is not active, the message is stored on an application server and is delivered to the receiver as soon as the receiver becomes alive. A single

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 447–457, 2011.

message can be sent to multiple receivers. In session-based messaging, both parties establish a session through an Invite request. After session establishment, users are then allowed to exchange messages and, at the end, the session is terminated through a Bye request. Immediate and session based messaging are described in figure 1 and 2 respectively.

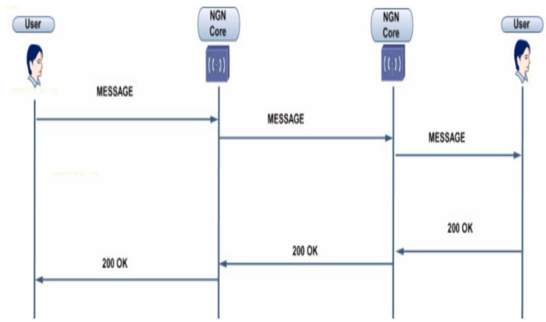


Fig. 1. Immediate messaging flow

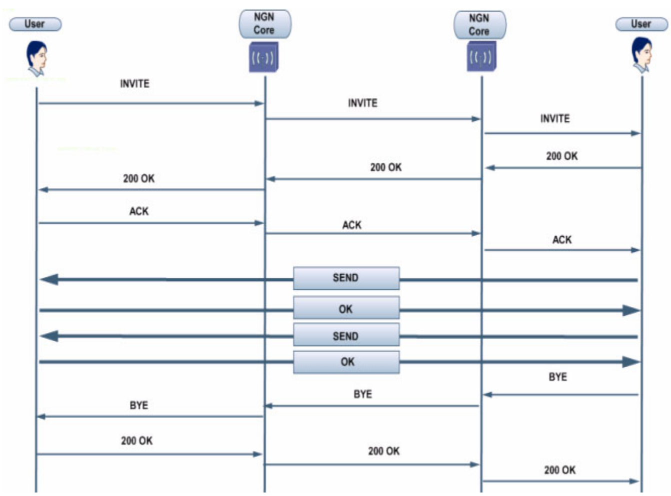


Fig. 2. Session based messaging flow

Day et al. in 2000 defines the terminologies used in presence and messaging and the minimum requirements those a protocol designed for presence and messaging should have [2, 3]. Milewski et al. in 2000 had presented the idea of live address book. [4]. 3GPP TS 22.940 defines the issues and terminologies of messaging service [5]. Conlan et al. in 2003 [6] discussed the current modeling and personalization techniques available in user specific services. According to Pailer [7] since Parlay APIs and SIP are declared standard by 3GPP so mapping of SIP over Parlay is a necessary step. Florian et al. in 2006 described that instead of subscribing to presence service individually, subscription should be allowed to a Resource List Server (RLS) [8].

Vishal et al. in 2006 presented a survey on security issues in presence and proposed few solutions to solve these security issues [9].

Pailer et al. in 2006 [10] proposed an architecture for location service enablers in IMS. Kranz et al. in 2006 [11] worked on instant messaging. They modified current instant messaging to ubiquitous presence system. Rosenberg in 2007 described that proper authorization should be implemented in presence service [12]. Jiang et al. in 2007 proposed a three layer for presence service. Layer one provides network services, layer two is responsible for basic services to all users and layer three contains the policies related to personalization service [13]. Salinas in 2007 described the advantages and disadvantages of using presence service [14]. Sedlar et al. in 2007 proposed the use of presence information in an enterprise environment [15]. Wang et al. in 2007 presented the idea of PoC session setup using rich presence information [16]. Beltern et al. in 2007 presented the fully distributed platform to deploy presence service [15].

Peternel et.al in 2008 describes the mechanism for enterprises to collect presence information from multiple presentities and to distribute among the consumers [17]. Chen et al. in 2009 argued that presence notifications put an extra load on network as well as on watcher. Therefore to reduce the load of presence notifications they worked on introducing a new notification method called as weakly consistent scheme [18]. Zheng et al. in 2009 presented a new mechanism to formulate instant messaging group based on their presence information [19]. Paolo et al. in 2009 argued that the major issue in success of presence service is the scalability [20,21]. Shafi et al. in 2009 described a three layer secure architecture for instant messaging. They proposed sender and receiver authorization as well as an IDP system at IMS core [22].

2 Problem Definition

IMS-based instant messaging is subject to various types of security threats. Details of these threats are given below:

2.1 Targeted Message Flooding

Sending enormous number of messages to a targeted user is possible through immediate messaging because it does not require any session establishment. So immediate messaging flood can acts as denial of service attack for a particular targeted user. Scenario is described in figure 3.

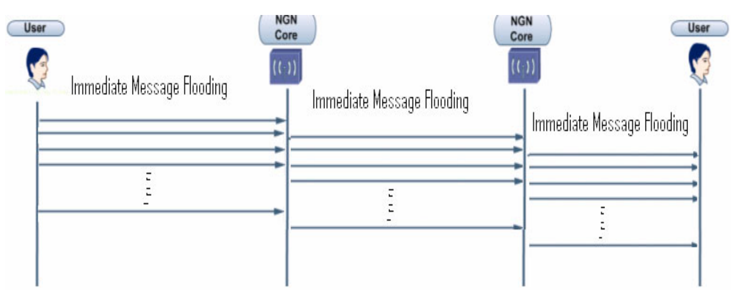


Fig. 3. Immediate message flooding on targeted user

2.1 Invite Flooding

Whenever a user wants to establish a session to exchange session based instant messaging, an invite request is routed towards the receiver. Sending too many invite requests may results in flooding and can cause denial of service. Figure 4 elaborate the problem to make it clearer.

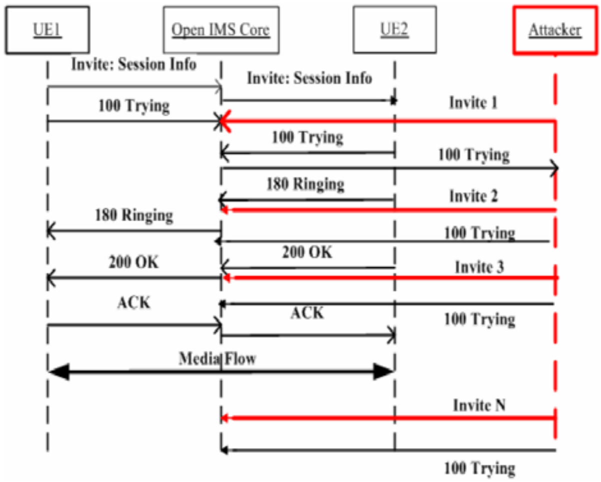


Fig. 4. Invite message flooding on targeted user

2.2 Session-Based Attacks

A session during the establishment process can be cancelled by sending a "Cancel" request. An attacker can cancel a half established connection between two communication parties. A "Re-invite" request is used to modify an established session. An attacker can change the session parameters by launching a session modification attack through a "Re-Invite" request. A "Bye" request is used to close a session, so an attacker by sending a "Bye" request can close a session without the permission of the communication parties. It is called session teardown attack.

3 Presence Enabled Proposed Secure Architecture: Part A

Presence service is proved as a good service provisioning mechanism when it is used with other services. Presence service inside the messaging service is also proposed. To use presence service users have to subscribe for it. Subscription and notification of presence service put an extra load on air interface of the network since it requires exchange of messages. In our solution there is no need to subscribe for the presence service. Presence service is checked and verify during the transit of instant message.

Whenever a user sends an immediate message or tries to initiate a session with particular user, the message or request is received by P-CSCF. P-CSCF after receiving the request forwards it towards S-CSCF. In the normal scenarios S-CSCF after performing necessary actions forwards the message or request towards the receiver's P-CSCF or towards particular application server. But in our solution S-CSCF, before forwarding a particular message or request towards the receiver, checks the presence information of sender as well as that of receiver. Checking the presence information of a sender is necessary because if someone is misusing (spoofing) the identity of the sender it can be caught from presence information of the sender. If the presence information of the sender contradicts with the sent request or message it means that the identity of the sender is misused by someone else. In that case message or request is discarded and sender is notified about the situation.

If the message or request does not contradict with the presence information of the sender then it is matched with the presence information of the receiver. Presence information of receiver may state that currently she is not willing to receive any instant message or currently she is in meeting and doesn't want to receive any message. In this case forwarding the request or message towards receiver not only results in disturbance at receiver's end but also results in wastage of valuable resources. So in our solution if the message or request contradicts with the receiver's presence information it is discarded immediately. Figures 5 and 6 describe the situation in more detail.

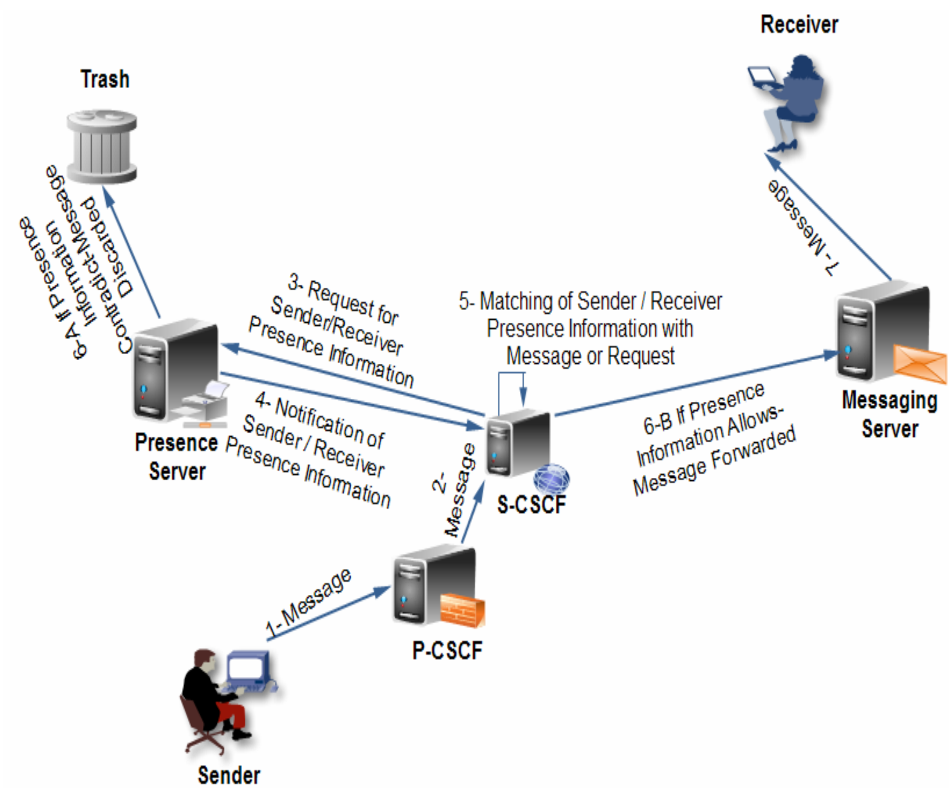


Fig. 5. Presence enabled instant messaging architecture

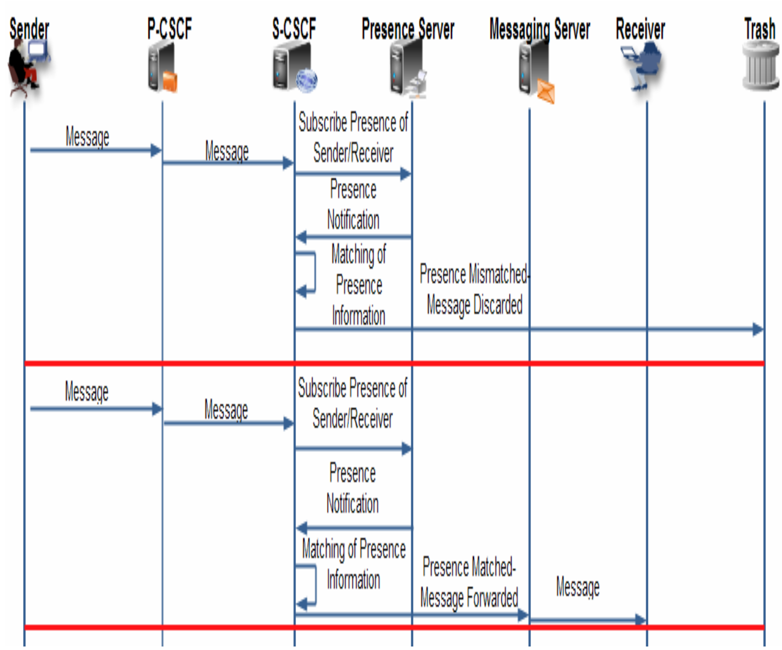


Fig. 6. Presence enabled instant messaging flow

3 Presence Enabled Proposed Secure Architecture: Part B

Checking the presence information of sender and receiver reduces the misuse of network resources and prevent the network from various types of attacks. But since presence information cannot describe all the current requirements of the users so blocking the attacker by using presence information only is not enough. Something more should be added to it to make the solution more secure. Authorization rules along with the presence information are added. Receiver specifies the authorization rules those are stored at messaging application server. Sender sends a message, S-CSCF first verifies the presence information of the sender to make it sure that the identity of the sender is not misuses. After that presence information of the receiver is checked to verify that what the current status of the receiver is. If the request or message does not contradict with the presence information of sender and receiver then it is forwarded towards messaging server. At messaging server user specify the receiving rules. These rules are more enrich as compared to presence information. Few examples of rules can be:

<?XML version=”1.0” encoding= “UTF-8”>
<Rules>
<Rule type=Sender_Authorization>
<User> 192.168.10.1 </User>

<Rule1>
<Name> User X </Name>
<time_start> 7:00 pm </Time_start>
<Time_end> 9:00 pm </Time_end>
<Action> Invite </Action>
<Alert> Not Allowed </Alert>
</Rule1>
</Rules>

These rules are compared with the incoming request or message and if any contradiction is found the request or message is discarded and the sender is notified. Figure 7 and 8 describe the scenario in detail.

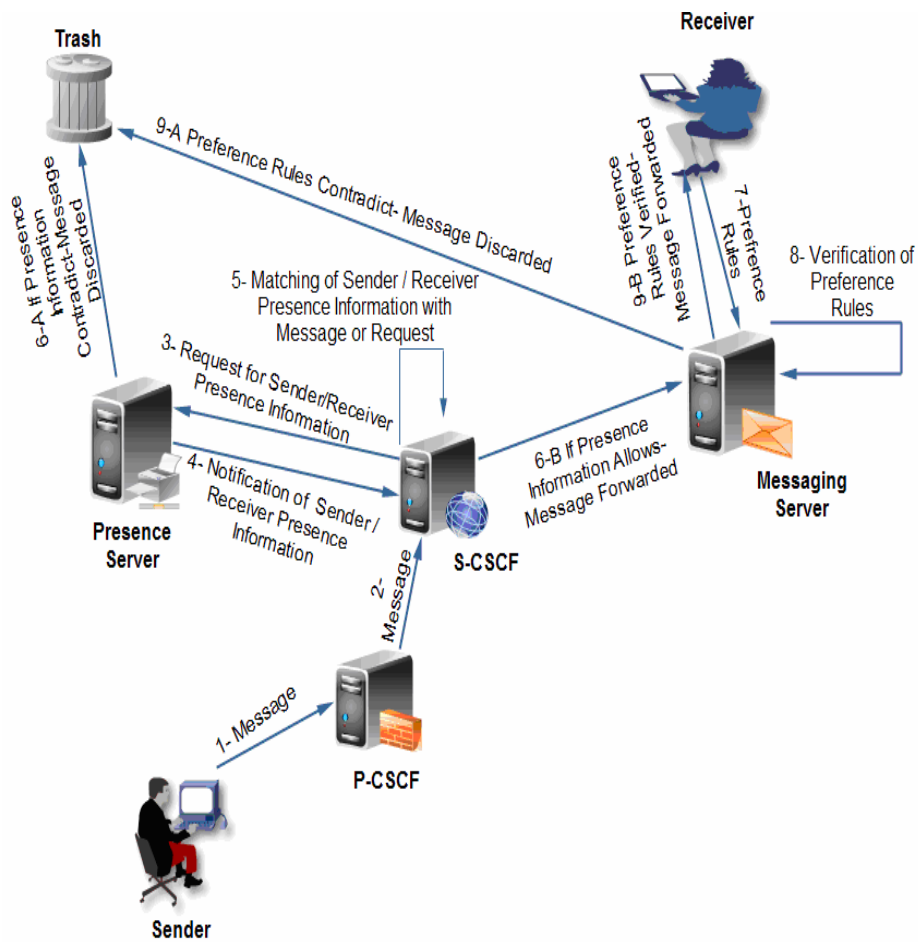


Fig. 7. Presence and preference based instant messaging architecture

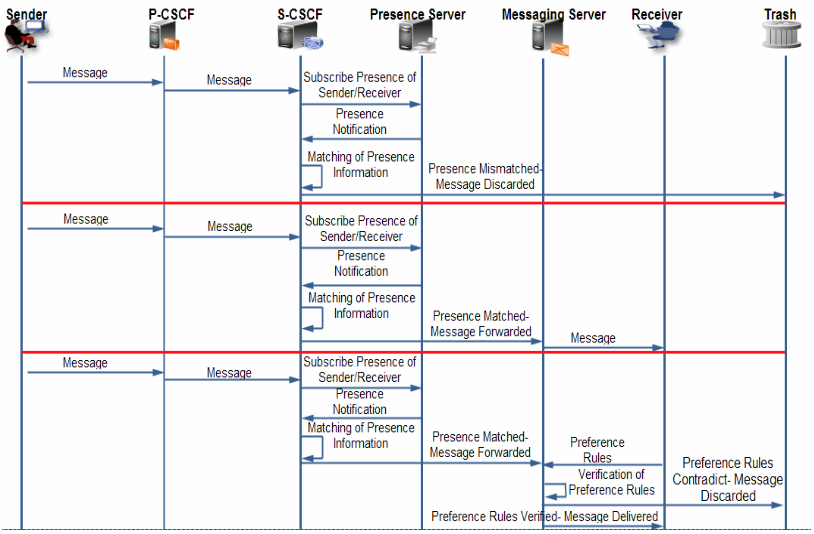


Fig. 8. Presence and preference based instant messaging flow

4 Results

Subscription and notification of presence information requires exchange of 4 messages over the air interface. Since our proposed solution checks the presence information during the transit so it is not necessary to subscribe for it. So it saves the 4 messages per subscription that results in saving the valuable resources and reduces to the load on air interface.

To test the validity of the proposed solution, four users are taken. Three users with IP addresses 192.168.10.1 and 192.168.10.2, 192.168.10.3 respectively, play the role of senders. They are named as “user A” and “user B,” and “Z” respectively. One receiver is configured with the IP addresses 192.168.10.4 and named as “Y”. Preferences of user Y are given below in the simple text.

1. Receive only 20 messages per hour and a message after every three minutes.
2. Give priority to A’s messages
3. B is allowed to send only 6 messages per hour and it has priority after A
4. Z is allowed to send 4 messages only.

First experiment tested validity of checking sender’s presence. For this purpose we set the presence information of user A as busy and it does not want to send and receive any instant message. Then we send an immediate message from user B by spoofing the identity of user A. We find that S-CSCF discards that message and a notification is received by user A that her ID is misused by some person. In the 2nd experiment we verify the presence information of receiver. User Y sets that currently she does not want to receive any message. After that user A sends a message and it is discarded by S-CSCF after checking the presence information of user Y.

In the third experiment we checked the validity of receiver's rules. User A sends 25 messages and user B sends 12 messages to Y. It is find that Y receives only 20 messages from user A. All other messages are discarded. This is because of rule 1 and rule 2. After that we send 10 messages from user A and 6 from user B and 4 messages from user Z. We find that all the messages are received by user Y.

In the 4th experiment we send 14 messages from user A, 25 messages from user B, and 10 messages from user Z. The experiment is conducted for one hour. Various scenarios are tested and Table 1 describes the results in detail.

Table 1. Results of 4th Experiment

Scenario	User A's delivered messages	User B's delivered messages	User Z's delivered messages
All the messages are forwarded during the first 2 minutes	14 (First)	6 (Last)	0
User Z and B forwarded the messages in first 2 minutes while the user A forwarded the messages after 20th minute	14 (Last)	6 (First)	0
User Z and B forwarded the messages in first 2 minutes while the user A forwarded the messages after 30th minute	10 (Last)	6 (First)	4 (Second)
User B forwarded the messages in first 2 minutes while the user Z forwarded the messages after 30th minute and User A after 55th minute	2 (Last)	6 (First)	4 (Second)
User B forwarded the messages in first 2 minutes while the user A and Z forwarded the messages after 30th minute	10 (Last)	6 (First)	0
User B forwarded the messages in first 5 minutes while the user A and X forwarded the messages after 55th minute	2 (Last)	6 (First)	0

Certainly it will take time to check the presence information of sender and receiver and to verify the rules specified by sender and receiver. We measure the delay caused by these solutions.

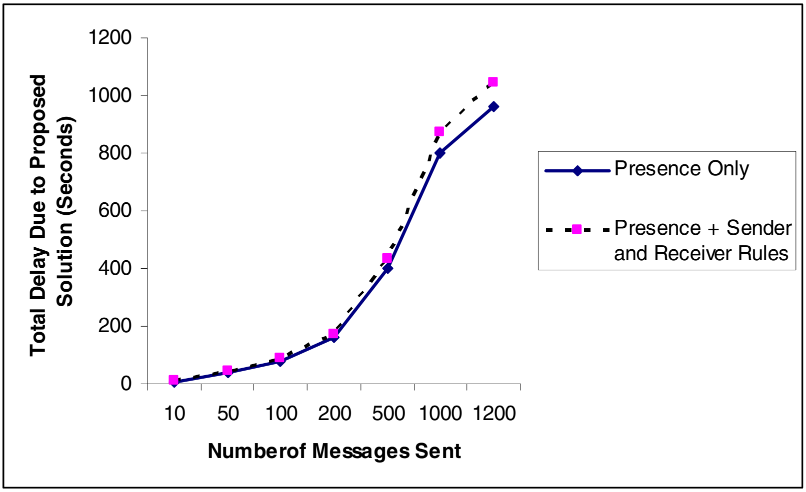


Fig. 9. Delay Calculations

First of all the solution minimize the loss incurred by a spoofed user. Since sender configures rules so these rules restricts the spoofed users to utilize sender's identity. Secondly targeted attacks on intended receivers are also minimized through configuring receiver's rule. Addition of presence information adds a new security level in IMS. Since in our proposed solution it is not required to subscribe the presence information so it reduced the overall delay. Moreover this mechanism also reduces the load from the air interface of the access network. In the figure 9 we have seen that delay of 0.8 seconds is introduced in a message due to checking the presence information and delay of 0.87 seconds is introduced by presence and rules of sender and receiver.

5 Conclusion and Future Work

We in this paper integrate the presence service with the instant messaging. We emphasized that users are not required to subscribe for the presence information and it will be checked dynamically during the transit. Sender's presence reduces the chances of spoofing and receiver's presence results in mitigating target attacks. Receiver also specifies the rules and these rules along with presence work to mitigate the targeted attacks as well as reduce the disturbance at receiver's end. In future we planned to integrate presence in other IMS based services like push to talk over cellular and multiparty conference.

References

[1] Poikselka, M., Mayer, G., Khartabil, H., Niemi, A.: IP multimedia concepts and services, 2nd edn. John Wiley & Sons Ltd., Chichester (2006)

[2] Day, M., Aggarwal, S., Mohr, G., Vincent, J.: A model for presence and instant messaging. Request for Comments: 2778 (2000)

[3] Day, M., Aggarwal, S., Mohr, G., Vincent, J.: Instant messaging/presence protocol requirements. Request for Comments: 2779 (2000)

[4] Milewski, A.E., Smith, T.M.: Providing presence cues to telephone users. In: CSCW 2000, Philadelphia, December 2-6 (2000)

[5] 3GPP TR 22.940 IP Multimedia Subsystem (IMS) messaging; Stage 1

[6] Conlan, O., Power, R., Barrett, K.: Next generation context aware adaptive services. In: Proceedings of the 1st International Symposium on Information and Communication Technologies, Dublin, Ireland (2003)

[7] Pailer, R., Stadler, J., Miladinovic, I.: Using PARLAY APIs over a sip system in a distributed service platform for carrier grade multimedia services. Wireless Networks 9, 353–363 (2003)

[8] Wegscheider, F.: Minimizing unnecessary notification traffic in the IMS presence system. In: 1st International Symposium on Wireless Pervasive Computing (2006)

[9] Singh, V.K., Schulzrinne, H.: A survey of security issues and solutions in presence, http://www1.cs.columbia.edu/~vs2140/presence/presencesecurity.pdf

[10] Pailer, R., Wegscheider, F., Bessler, S.: A terminal-based location service enabler for the IP multimedia subsystem. In: Proceedings of WCNC 2006 (2006)

[11] Kranz, M., Holleis, P., Schmidt, A.: Ubiquitous presence systems. In: Proceedings of SAC 2006, Dijon, France, April 23-27 (2006)

[12] Rosenberg, J.: Presence authorization rules, Request for Comments: 5025 (2007)

[13] Jiang, D., Yeap, T.H., Logrippo, L., Liscano, R.: Personalization for SIP multimedia communications with presence, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.63.8748

[14] Salinas, A.: Advantages and disadvantages of using presence service, http://www.tml.tkk.fi/Publications/C/21/salinas_ready.pdf

[15] Sedlar, U., Bodnaruk, D., Zebec, L., Kos, A.: Using aggregated presence information in an enterprise environment, https://www.icin.biz/files/programmes/Poster-7.pdf

[16] Wang, L., Gu, D.: A study on session setup for group communications in push-to-talk over cellular using rich presence, http://ieeexplore.ieee.org/iel5/4391971/4391972/04392055.pdf?arnumber=4392055

[17] Peternel, K., Zebec, L., Kos, A.: Using presence information for an effective collaboration. In: 6th International Symposium on Communication Systems, Networks and Digital Signal Processing (2008)

[18] Chen, W.-E., Lin, Y.-B., Liou, R.-H.: A weakly consistent scheme for IMS presence service. IEEE Transactions on Wireless Communications 8(7) (2009)

[19] Zheng, W., Zhang, C., Yu, C., Yang, P., Mu, Y., Zhang, X.: Design of presence based dynamic group IM in IMS. In: Proceedings of the 2009 Fifth International Joint Conference on INC, IMS and IDC (2009)

[20] Bellavista, P., Corradi, A., Foschini, L.: IMS-based presence service with enhanced scalability and guaranteed QoS for interdomain enterprise mobility. IEEE Wireless Communications 16(3) (2009)

[21] Bellavista, P., Corradi, A., Foschini, L.: Enhancing the scalability of IMS-based presence service for LBS applications. In: Proceedings of the 2009 33rd Annual IEEE International Computer Software and Applications (2009)

[22] Khan, Z.S., Sher, M., Rashid, K., Rahim, A.: A three layer se-cure architecture for IP multimedia subsystem based instant messaging. Information Security Journal: A Global Prespective 18, 1–10 (2009)

Two-Directional Two-Dimensional Random Projection and Its Variations for Face and Palmprint Recognition

Lu Leng[1], Jiashu Zhang[1], Gao Chen[1],
Muhammad Khurram Khan[2], and Khaled Alghathbar[2,3]

[1] Sichuan Province Key Lab of Signal and Information Processing,
Southwest Jiaotong University, Chengdu Sichuan 610031, P.R. China
lenglu@mars.swjtu.edu.cn, jszhang@home.swjtu.edu.cn
[2] Center of Excellence in Information Assurance,
King Saud University, Riyadh, Saudi Arabia
mkhurram@ksu.edu.sa
[3] Information Systems Department, College of Computer and Information Sciences,
King Saud University, Riyadh, Saudi Arabia

Abstract. 2DRP (two-dimensional random projection) is two-dimensional extension of one-dimensional RP (random projection) to keep biometric images from being reshaped to vectors before RP for recognition. We propose a novel method called $(2D)^2$RP (two-directional two-dimensional random projection) for feature extraction of biometrics. $(2D)^2$RP directly projects the image matrix from high-dimensional space to low-dimensional space to extract optimal projective vectors at row-direction and column-direction. $(2D)^2$RP, similar to RP, can also avoid the problems of singularity, SSS (small sample size) and over-fitting; furthermore it has much less storage and computational cost than RP. Besides, the variations of $(2D)^2$RP combined with 2DPCA and 2DLDA are developed. Experimental results and comparison discussion among $(2D)^2$RP and its variations on face and palmprint databases confirm the performance and effectiveness of $(2D)^2$RP and its variations.

Keywords: $(2D)^2$RP, Principal component analysis, Linear discriminant analysis, Face recognition, Palmprint recognition.

1 Introduction

Biometric-based methods are ideal and secure for identity authentication [1-2]. The techniques of dimensionality reduction, widely used for biometric recognition, transform high dimensional data into meaningful low dimensional representation. PCA (principal component analysis) [3] and LDA (linear discriminant analysis) [4] are two popular methods widely used to reduce dimension for biometric recognition. RP (random projection) is a simple, easy and fast technique of dimensionality reduction. The theory basis or RP can be referred Johnson-Lindenstrauss lemma [5]. RP is independent to training data and avoids the problems of singularity, SSS (small sample size) and over-fitting [6-7]; hence it has also been widely used for biometric recognition. However, biometric images must be reshaped to vectors before RP for recognition. 2DRP,

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 458–470, 2011.

two-dimensional extension of one-dimensional RP, addresses the above problem [8]. PCA and LDA have the similar extension progress. PCA is extended to 2DPCA [9] and $(2D)^2$PCA [10]; LDA is extended to 2DLDA [11] and $(2D)^2$LDA [12].

We propose a novel method called $(2D)^2$RP (two-directional two-dimensional random projection) for feature extraction of biometrics. $(2D)^2$RP directly projects the image matrix from high-dimensional space to low-dimensional space at row-direction and column-direction. $(2D)^2$RP not only has the similar strengths of RP, but also reduces storage and computational cost greatly. Since PCA and LDA have some advantages that don't belong to RP, so the methods fusing RP and PCA/LDA can play complementary advantages. The variations of $(2D)^2$RP combining 2DRP with 2DPCA or 2DLDA are developed and compared. The sparse random matrix for random projection used in this paper is generated according to [13]. Since face and palmprint are two popular biometrics for identity authentication [14-15], the experimental results on ORL face database, Yale face database and PolyU palmprint database confirm the performance and effectiveness of $(2D)^2$RP and its variations. The comparisons among the methods are discussed and analyzed extensively.

The rest of the paper is organized as follows: Section 2 briefly reviews sparse RP, 2DPR and presents $(2D)^2$RP. Section 3 elaborates four variations of $(2D)^2$RP. Section 4 demonstrates the experimental results and discussions. Finally we draw our conclusions in Section 5.

2 $(2D)^2$RP

2.1 Sparse RP

RP is an algorithm for data-independent dimension reduction. Assume an image $\mathbf{X} \in \mathbb{R}^{m \times n}$, the random matrix to project $\mathbf{X}$ onto k-dimensional ($k<mn$) subspace can be computed by

$$\mathbf{y}_{k\times 1} = \mathbf{R}_{k\times mn} \cdot \mathbf{x}_{mn\times 1} \tag{1}$$

$\mathbf{X}$ is reshaped to $\mathbf{x}_{mn\times 1}$ that is a vector. The elements r_{ij} of traditional random matrix $\mathbf{R}$ are often Gaussian distributed. In fact, the random matrix with Gaussian distribution can be replaced by a sparse random matrix as follows.

$$\Pr(r_{ij} = t) = \begin{cases} 0.5, t = 1 \\ 0.5, t = -1 \end{cases} \tag{2}$$

2.2 2DRP

Images must be reshaped to vectors before RP for recognition. 2DRP, two-dimensional extension of one-dimensional RP, addresses the above problem. 2DRP directly projects the image matrix $\mathbf{X}$ from high-dimensional space to low-dimensional space $\mathbf{Y}$ at row-direction by

$$\mathbf{Y}_{m\times d} = \mathbf{X}_{m\times n} \mathbf{T}_{n\times d} \tag{3}$$

or column-direction by

$$\mathbf{Y}_{b\times n} = \mathbf{R}_{b\times m}\mathbf{X}_{m\times n} \tag{4}$$

where $b<m$ and $d<n$. $\mathbf{R}$ and $\mathbf{T}$ are left and right random projection matrices, respectively.

2.3 $(2D)^2$RP

Although 2DRP has much less storage and computational cost, it requires much more coefficients for image representation than one-dimensional RP approaches. In fact, 2DRP can perform in two directions at the same time. $(2D)^2$RP directly projects the image matrix from high-dimensional space to low-dimensional space at row-direction and column-direction as

$$\mathbf{Y}_{b\times d} = \mathbf{R}_{b\times m}\mathbf{X}_{m\times n}\mathbf{T}_{n\times d} \tag{5}$$

$\mathbf{R}$ and $\mathbf{T}$ denote the left mapping matrix for column-direction dimension reduction and right mapping matrix for row-direction dimension reduction, respectively. Equation (5) can be rewritten as

$$\mathbf{S} = \mathbf{T}^T \otimes \mathbf{R} \in \mathbb{R}^{bd\times mn} \tag{6}$$

Assume $vec(\mathbf{X})$ is the vectorization of the matrix $\mathbf{X}$ formed by stacking the columns of $\mathbf{X}$ into a single column vector, then

$$vec(\mathbf{RXT}) = \mathbf{S}\cdot vec(\mathbf{X}) \tag{7}$$

The element s_{pq} in $\mathbf{S}$ can be computed by

$$s_{pq} = r_{ij}\cdot t_{rs} \tag{8}$$

If r_{ij} and t_{rs} are mutual independent, then

$$\begin{aligned}
&\Pr(s_{pq}=1)\\
&=\Pr(r_{ij}\cdot t_{rs}=1)\\
&=\Pr(r_{ij}=1,t_{rs}=1)+\Pr(r_{ij}=-1,t_{rs}=-1)\\
&=\Pr(r_{ij}=1)\cdot\Pr(t_{rs}=1)+\Pr(r_{ij}=-1)\cdot\Pr(t_{rs}=-1)\\
&=0.5\times0.5+0.5\times0.5=0.5\\
&\Pr(s_{pq}=-1)\\
&=\Pr(r_{ij}\cdot t_{rs}=-1)\\
&=\Pr(r_{ij}=1,t_{rs}=-1)+\Pr(r_{ij}=-1,t_{rs}=1)\\
&=\Pr(r_{ij}=1)\cdot\Pr(t_{rs}=-1)+\Pr(r_{ij}=-1)\cdot\Pr(t_{rs}=1)\\
&=0.5\times0.5+0.5\times0.5=0.5
\end{aligned}$$

It is obvious that 1D random projection matrix $\mathbf{S}$ satisfies the distribution in (2) if random matrices $\mathbf{R}$ and $\mathbf{T}$ are generated according to distribution in (2), respectively.

3 Variations of $(2D)^2$RP

PCA and LDA have some advantages that don't belong to RP, so the methods fusing RP and PCA/LDA can play complementary advantages. Four variations of $(2D)^2$RP combining 2DRP with 2DPCA or 2DLDA are developed. The equations of four variations, namely $(2D)^2$RPPCA, $(2D)^2$PCARP, $(2D)^2$RPLDA and $(2D)^2$LDARP, are (9)~(12), respectively. The variations project image matrix from high-dimensional space **X** to low-dimensional space **Y** at row-direction and column-direction with different methods by **B** and **A**, respectively, where **A** and **B** denote the left mapping matrix for column-direction dimension reduction and right mapping matrix for row-direction dimension reduction. The superscript of **A** or **B** indicates the method that generates the mapping matrix **A** or **B**. The methods of two-directional two-dimensional dimension reduction are shown in Table 1.

$$\mathbf{Y}_{b\times d} = \mathbf{A}_{b\times m}^{RP}\mathbf{X}_{m\times n}\mathbf{B}_{n\times d}^{PCA} \tag{9}$$

$$\mathbf{Y}_{b\times d} = \mathbf{A}_{b\times m}^{PCA}\mathbf{X}_{m\times n}\mathbf{B}_{n\times d}^{RP} \tag{10}$$

$$\mathbf{Y}_{b\times d} = \mathbf{A}_{b\times m}^{RP}\mathbf{X}_{m\times n}\mathbf{B}_{n\times d}^{LDA} \tag{11}$$

$$\mathbf{Y}_{b\times d} = \mathbf{A}_{b\times m}^{LDA}\mathbf{X}_{m\times n}\mathbf{B}_{n\times d}^{RP} \tag{12}$$

Table 1. Two-directional two-dimensional dimension reduction

Variations	Column –direction Dimension reduction	Row–direction Dimension reduction
$(2D)^2$RP	RP	RP
$(2D)^2$PCA	PCA	PCA
$(2D)^2$LDA	LDA	LDA
$(2D)^2$RPPCA	RP	PCA
$(2D)^2$PCARP	PCA	RP
$(2D)^2$RPLDA	RP	LDA
$(2D)^2$LDARP	LDA	RP

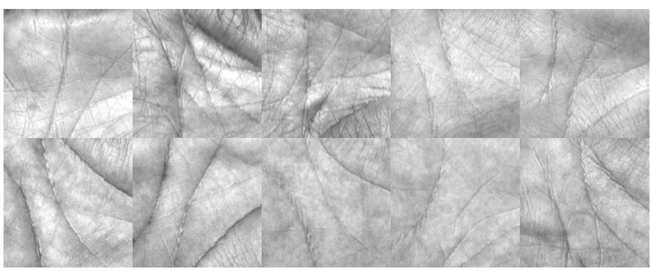
4 Experimental Results and Discussions

4.1 Experimental Details

In order to evaluate the proposed method, the experiments are performed on ORL face database, Yale face database and PolyU palmprint database. The samples of the databases tested in this paper are shown in Fig. 1.

(a) ORL

(b) Yale



(c) PolyU

Fig. 1. The samples of the face and palmprint databases

The details of the testing databases are listed in Table 2. The six data of each database are the number of classes, the number of samples for each class, the number of samples for training for each class, the number of samples for testing for each class, the size of original images and the down-sampled size, respectively. All simulations split the datasets into train and test dataset randomly and the results are the average of the numerous runs.

Table 2. Database and simulations details

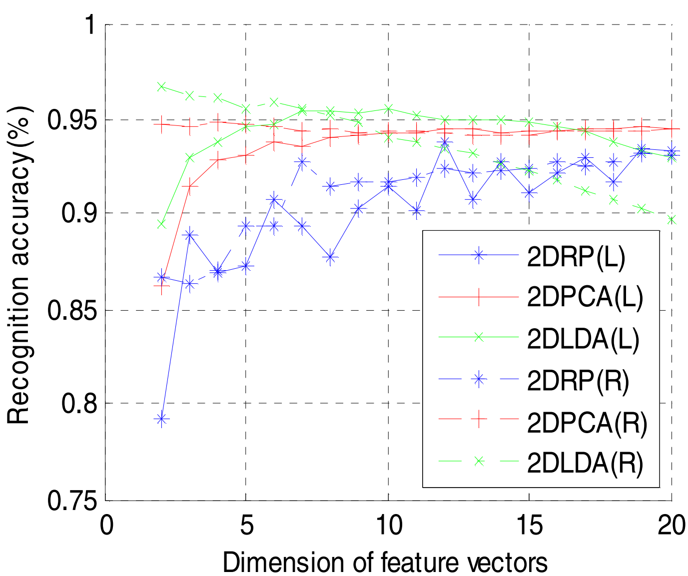
Database	ORL	Yale	PolyU
Number of classes	40	15	100
Samples per class	10	11	6
Train number per class	5	6	3
Test number per class	5	5	3
Image size	112×92	320×240	128×128
Down-sampled size	56×46	80×60	64×64

4.2 2D Dimension Reduction

Fig. 2 shows the comparison of recognition accuracies of 2DPCA, 2DLDA and 2DRP. (L) and (R) denote the left mapping matrix for column-direction dimension reduction and right mapping matrix for row-direction dimension reduction, respectively. Assume the size of the image is $m \times n$. The dimension b of feature matrices in the group (L) denotes the height of the feature matrices, that is, the size of feature matrices are $b \times n$; while the dimension d of feature matrices in the group (R) denotes the width of the feature matrices, that is, the sizes of feature matrices are $m \times d$.

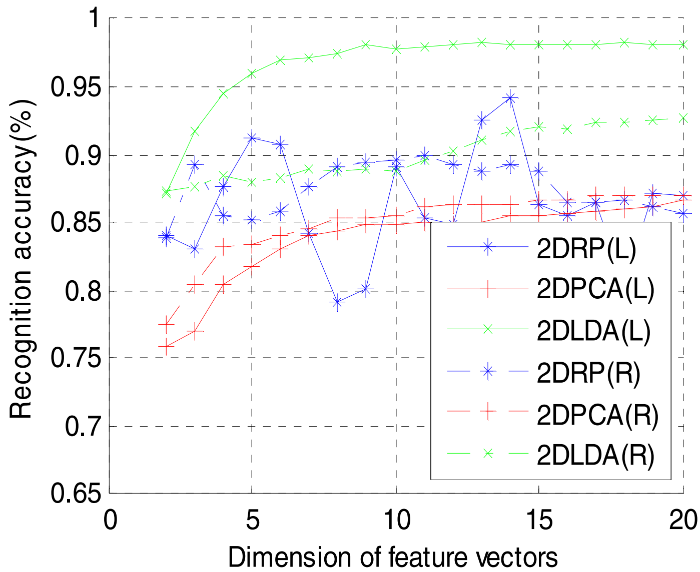
From the experimental results of Fig. 2, we can see:

(1) 2DRP(R) is more stable than 2DRP(L). 2DRP(L) has better recognition accuracy than 2DRP(R) on PolyU database, but it is hard to compare the performance between 2DRP(L) and 2DRP(R) on face databases. The performance of 2DRP is less stable than 2DPCA and 2DLDA with the change of dimension of feature vectors.

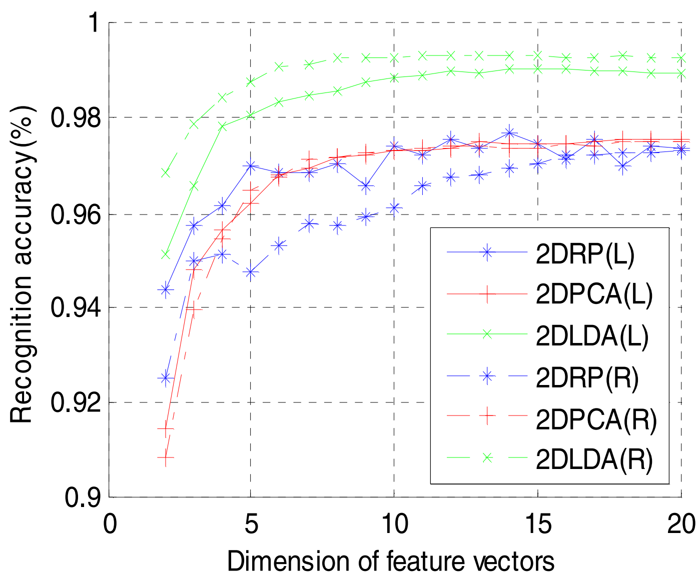


(a) ORL

Fig. 2. The comparison of recognition accuracy among 2DPCA, 2DLDA and 2DRP



(b) Yale



(c) PolyU

Fig. 2. *(continued)*

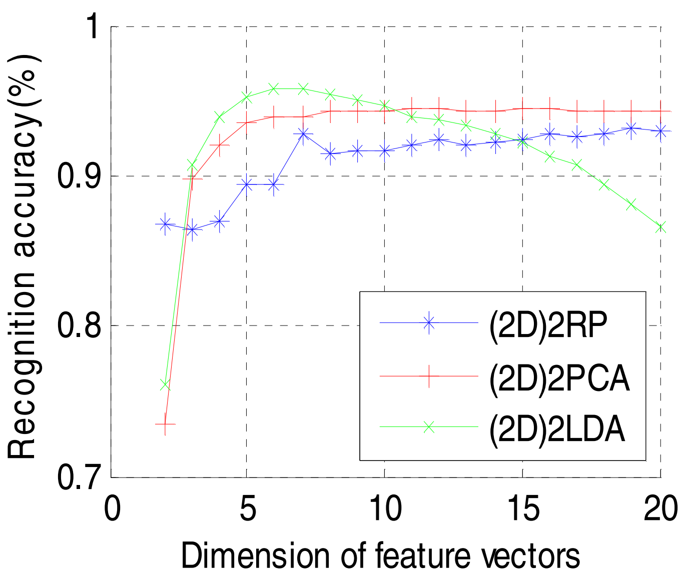
(2) The performance of 2DPCA(L) and 2DPCA(R) is approximate on all databases.
(3) The performance of 2DLDA(R) degrades with the increase of dimension of feature vectors on ORL database. 2DLDA(L) is better than 2DLDA(R) on Yale database, but in contrast, 2DLDA(R) outperforms 2DLDA(L) on Yale database.
(4) As a whole, 2DLDA is better than 2DPCA and 2DRP on Yale and Palmprint databases. It is hard do compare the performance between 2DPCA and 2DLDA on ORL database.

4.3 $(2D)^2$ Dimension Reduction

Fig. 3 shows the comparison of recognition accuracies of $(2D)^2$PCA, $(2D)^2$LDA and $(2D)^2$RP. The dimension d of feature matrices is the height and the width of the feature matrices, that is, the sizes of feature matrices are $d \times d$.

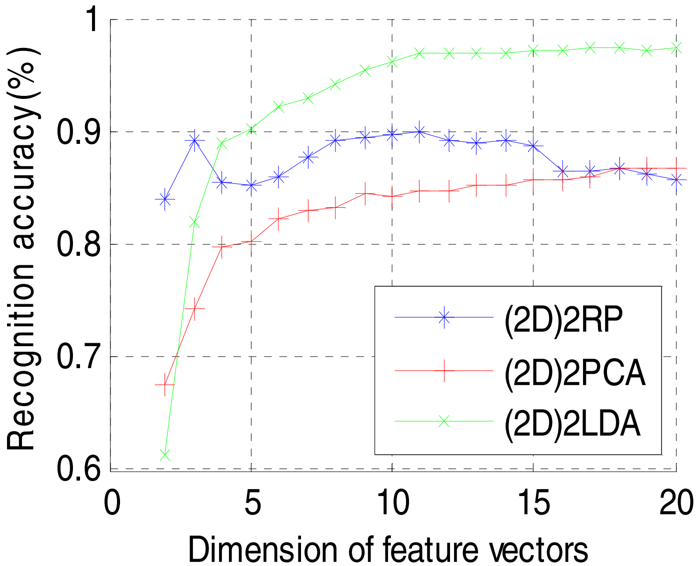
From the experimental results of Fig. 3, we can see:

(1) As a whole, $(2D)^2$LDA has better performance than $(2D)^2$PCA and $(2D)^2$RP on the three databases, but its performance degrades when the dimension of the feature vectors is large on ORL database.
(2) $(2D)^2$PCA outperforms $(2D)^2$RP on ORL database; while $(2D)^2$RP outperforms $(2D)^2$PCA on Yale database. $(2D)^2$PCA and $(2D)^2$RP have similar performance on PolyU database.

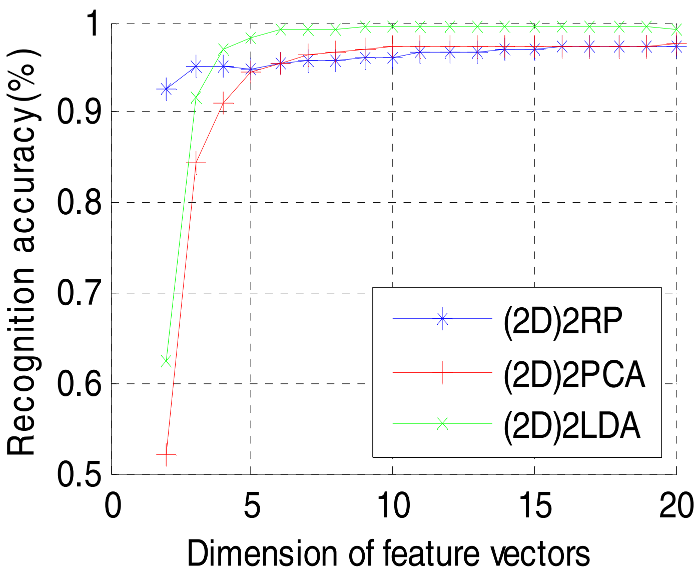


(a) ORL

Fig. 3. The comparison of recognition accuracy among $(2D)^2$PCA, $(2D)^2$LDA and $(2D)^2$RP



(b) Yale

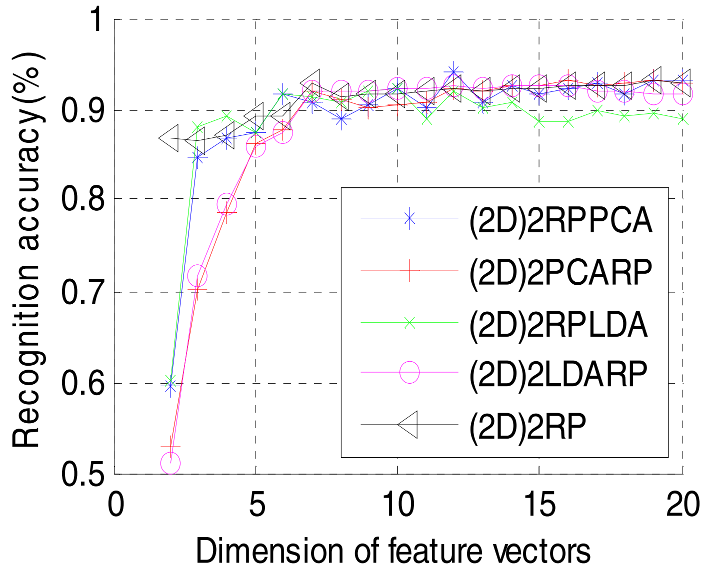


(c) PolyU

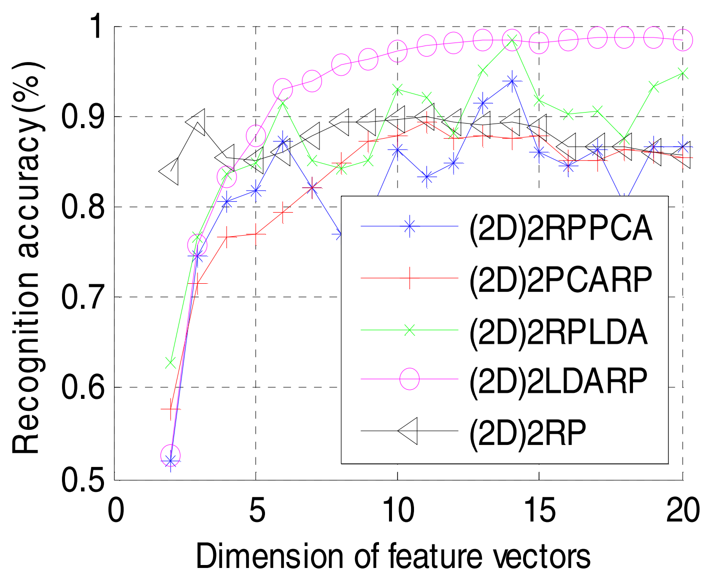
Fig. 3. *(continued)*

4.4 Variations of $(2D)^2$RP

Fig. 4 shows the comparison of recognition accuracies of $(2D)^2$RP, $(2D)^2$RPPCA, $(2D)^2$PCARP, $(2D)^2$RPLDA and $(2D)^2$LDARP. The dimension d of feature matrices is the height and the width of the feature matrices, that is, the sizes of feature matrices are d×d.

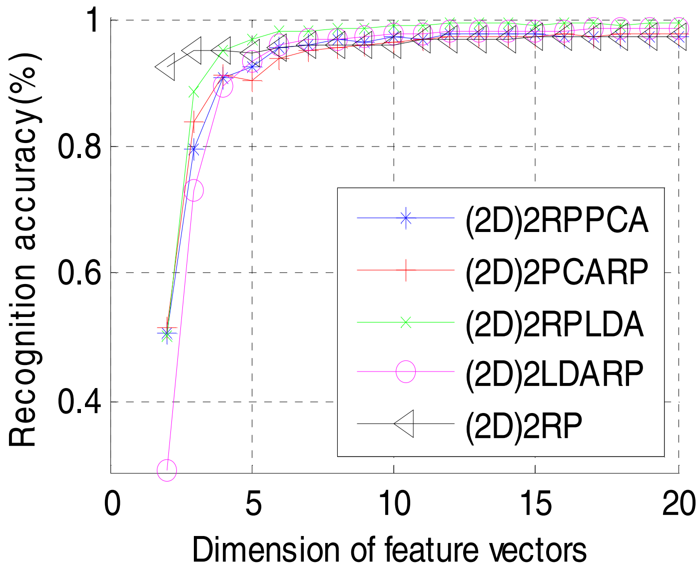


(a) ORL



(b) Yale

Fig. 4. The comparison of recognition accuracy among $(2D)^2$RP, $(2D)^2$RPPCA, $(2D)^2$PCARP, $(2D)^2$RPLDA and $(2D)^2$LDARP



(c) PolyU

Fig. 4. *(continued)*

From the experimental results of Fig. 4, we can see:

(1) Although $(2D)^2$RP has not the best recognition accuracy, it has the highest recognition accuracy when the dimension of feature vectors is very little. $(2D)^2$RP has the strongest compression ability property when the compression ratio is large.
(2) All methods have desired performance under the condition of little variation of rotation, scaling and titling (on ORL and PolyU databases). However, $(2D)^2$LDARP has better recognition accuracy under the condition of varied facial expression and lighting configuration (on Yale database).
(3) $(2D)^2$LDARP that combines the advantages of RP and LDA has the desired performance and is more stable than other methods on face databases. The other methods are more fluctuant than $(2D)^2$LDARP on face databases.
(4) All methods have more stable recognition accuracies on palmprint database than those on face databases.

5 Conclusions

$(2D)^2$RP is proposed as a novel and efficient face and palmprint recognition method which has desired representation and recognition ability even when the compression ratio is large. $(2D)^2$RP directly projects the image matrix from high-dimensional space to low-dimensional space at row-direction and column-direction, so it keeps biometric images from being reshaped to vectors before RP for recognition. $(2D)^2$RP has the similar strengths of RP with much less storage and computational cost.

Besides, the variations of $(2D)^2$RP combining RP with 2DPCA or 2DLDA, including $(2D)^2$RPPCA, $(2D)^2$PCARP, $(2D)^2$RPLDA and $(2D)^2$LDARP, play complementary advantages of RP and PCA or LDA. The comparison among $(2D)^2$RP and its variations was discussed and analyzed extensively. The variations not only have strengths of RP, but also combine the optimal reconstruction information from 2DPCA or the optimal discriminative information from 2DLDA.

Acknowledgments. The authors would like to express their sincere thanks to the Biometric Research Center at the Hong Kong Polytechnic University for providing us the Palmprint Database. The authors would also like to thank the associate editor and anonymous reviewers for their comments, which significantly helped to improve this paper.

This work was partially supported by National Science Foundation of P. R. China (Grant: 60971104), Research Fund for the Doctoral Program of Higher Education of China (Grant: 20090184110008), Sichuan Youth Science & Technology Foundation (Grant: 09ZQ026-091), Open Research Foundation of Chongqing Key Laboratory of Signal and Information Processing, Chongqing University of Posts and Telecommunications (Grant: CQSIP-2009-01), Fundamental Research Funds for the Central Universities (Grant: SWJTU09ZT16), and Science & Technology Key Plan Project of Chengdu (Grant: 10GGYB649GX-023).

References

1. Khan, M.K., Xie, L., Zhang, J.S.: Chaos and NDFT-based Concealing of Fingerprint-biometric Data into Audio Signals for Trustworthy Person Authentication. Dig. Sign. Proc. 20, 179–190 (2010)
2. Leng, L., Zhang, J.S., Khan, M.K., Chen, X., Ji, M., Alghathbar, K.: Cancelable PalmCode Generated from Randomized Gabor Filters for Palmprint Template Protection. Sci. Res. Ess. 6, 784–792 (2011)
3. Turk, M., Pentland, A.: Eigenfaces for Recognition. J. Cogn. Neur. 3, 71–86 (1991)
4. Belhumeour, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. IEEE Trans. Patt. Anal. Mach. Int. 19, 711–720 (1997)
5. Johnson, W., Linderstrauss, J.: Extensions of Lipshitz Mapping into Hilbert Space. Cont. Math. 26, 189–206 (1984)
6. Majumdar, A., Ward, R.K.: Robust Classifiers for Data Reduced via Random Projections. IEEE Trans. Syst. Man Cyb. Part B: Cyb. 40, 1359–1371 (2010)
7. Choi, K., Toh, K., Byun, H.: Realtime Training on Mobile Devices for Face Recognition Applications. Patt. Recogn. 44, 386–400 (2011)
8. Sanguansat, P.: Two-Dimensional Random Projection for Face Recognition. In: 1st Int. Conf. Perv. Comp. Sign. Proc. Appl. (PCSPA), pp. 1107–1110. IEEE Press, Harbin (2010)
9. Yang, J., Zhang, D., Frangi, A.F., Yang, J.Y.: Two-dimensional PCA: a New Approach to Appearance-based Face Representation and Recognition. IEEE Trans. Patt. Anal. Mach. Int. 26, 131–137 (2004)
10. Zhang, D.Q., Zhou, Z.H.: $(2D)^2$PCA: Two-directional Two-dimensional PCA for Efficient Face Representation and Recognition. Neurocomputing 69, 224–231 (2005)

11. Li, M., Yuan, B.Z.: 2D-LDA: A Statistical Linear Discriminant Analysis for Image Matrix. Patt. Recogn. Lett. 26, 527–532 (2005)
12. Noushath, S., Kumar, G.H., Shivakumara, P.: $(2D)^2$LDA: An Efficient Approach for Face Recognition. Patt. Recogn. 39, 1396–1400 (2006)
13. Achlioptas, D.: Database-friendly Random Projections: Johnson-Lindenstrauss with Binary Coins. J. Comp. Syst. Sci. 66, 671–687 (2003)
14. Leng, L., Zhang, J.S., Khan, M.K., Chen, X., Alghathbar, K.: Dynamic Weighted Discrimination Power Analysis: a Novel Approach for Face and Palmprint Recognition in DCT Domain. Int. J. Phys. Sci. 5, 2543–2554 (2010)
15. Leng, L., Zhang, J.S., Xu, J., Khan, M.K., Alghathbar, K.: Dynamic Weighted Discrimination Power Analysis in DCT Domain for Face and Palmprint Recognition. In: Int. Conf. Inf. Comm. Techn. Conv. (ICTC), pp. 467–471. IEEE Press, Jeju (2010)

A Framework for Intelligent Tutoring in Collaborative Learning Systems Using Service-Oriented Architecture

Fang-Fang Chua and Chien-Sing Lee

Faculty of Information Technology, Multimedia University,
Jalan Multimedia, 63100 Cyberjaya, Selangor, Malaysia
{ffchua,cslee}@mmu.edu.my

Abstract. Artificial Intelligence techniques are applied in learning systems to enhance the quality of interaction between the users and the system. E-learning system components, learning services and learning companion services have been implemented in a traditional manner whereby there is little possibility of reuse due to the tight coupling of components and lack of reuse for learning activities. The scope of our research aims to design a framework for intelligent tutoring in collaborative learning systems using Service-oriented Architecture. We aim to create a better personalized learning environment whereby users can interact with the system according to their needs and levels. Teaching strategy is designed to provide suitable assistance according to different user groups.

Keywords: Collaborative, Service-oriented Architecture, Reuse, Tutoring.

1 Introduction

E-learning is the application of computer-based technologies to enhance teaching and learning [1]. [2] describes E-learning as a networked activity, which makes it capable of instant updating, storage or retrieval, distribution and sharing of instruction or information. In this information age, the educational paradigm has shifted from the traditional transmissive classroom to go online. The transmissive paradigm emphasizes on the transfer of knowledge from lecturer to student. Hence, it may not be conducive as active learning occurs when students can take a pro-active role in questioning, sharing ideas and applying prior knowledge to develop new ideas.

Collaboration has been shown to be an effective medium that exposes students to diverse and rich perspectives on thinking, creating cognitive dissonance and stimulating reflection, assimilation and accommodation of knowledge [3] and [4]. Easy coordination of collaborative activities is provided through the incorporation of the cognitive apprenticeship model.

In computer-aided instruction, scaffolds are needed to further enhance the quality of interaction and learning effectiveness among students and between the students and the interactive system. An Intelligent Tutoring System (ITS), broadly defined, is any computer system that provides direct customized instruction or feedback to students, i.e. without the intervention of human beings [5]. It consists of the Domain Model, Student Model and Pedagogical model and focuses on applying artificial intelligence techniques and agent technology to facilitate the learning process. Due to distributed

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 471–484, 2011.

resources and learning components everywhere, Service-Oriented Architecture (SOA) plays an important role to enable reuse of components, pedagogy and learning activities in this heterogeneous environment. Users or learners can then discover and compose the required services anytime anywhere. Distributed interactivity which includes learning activities and personalized coaching through Web services enhances the interoperability and reusability of e-learning content in a collaborative environment.

2 Background to the Study

In our earlier work, we addressed the need to identify users' perception towards collaborative learning using concept maps within a common knowledge space [6], and the need to identify how to guide users meaningfully during collaborative learning [7]. Pilot test findings indicate that all respondents agreed that the Merlin collaborative concept mapping whiteboard enabled them to externalize their knowledge and represent the concepts and ideas easily. Furthermore, setting personal and common goals as well as modelling helped create positive learning experiences. Qualitative data showed that most of the participants achieved personal and common learning goals by using Merlin in completing their assignments.

Subsequently, [7] enhanced the coaching method in the Merlin collaborative learning environment, in an effort to improve the interactivity process. By reviewing existing learning systems on collaborative learning and intelligent tutoring, improvements are proposed in Merlin version 3 to increase the degree of coaching and advice to be provided to students. Merlin version 3 aims to improve the mapping between the adaptation and pedagogical modules in ITSs, which lead to more effective adaptation of tutoring techniques. It encourages real-time collaboration in concept mapping learning environments. It also encourages correlation between individual and collaborative workspace. The adoption of coaching and learning companion strategies will enhance the hinting strategies.

Next, SOA is utilized in an effort to identify a framework for creating reusable learning services in a distributed collaborative environment. Service-Oriented Computing (SOC) incorporates a new computing paradigm of publish discover- compose, in which service providers publish services, service dynamically consumers discover right service and aggregate them into a service composition. [8] presented a generic SOC process model from which they identified key artifacts in Service-Oriented development; *Application Requirements, Business Process, Service Composition, and Service Component*. A framework for verifying key artifacts of service-oriented development by defining a foundation for the verification framework; a generic process model of SOC and its key artifacts were presented. Verification paths and criteria for service-oriented verification were defined and based on the criteria, methods for verifying the completeness and correctness of SOC artifacts were proposed. A case study was applied to show the applicability of the framework in practice.

2.1 Scope of the Study

In this paper, we extend from the earlier works to investigate how to facilitate intelligent tutoring in a distributed and reusable environment. Our proposed Merlin version 3 system architecture which forms the platform for investigation is shown in Figure 1.

Merlin version 3 consists of student model, pedagogic model and expert model. The Parser Layer determines the students' interaction activities. The Dialogue Agent will be the communication tool for the students' interactions where new data, which are the statements from the students, will be parsed to the student model and all the student files and activities will be stored in the Dialogue History.

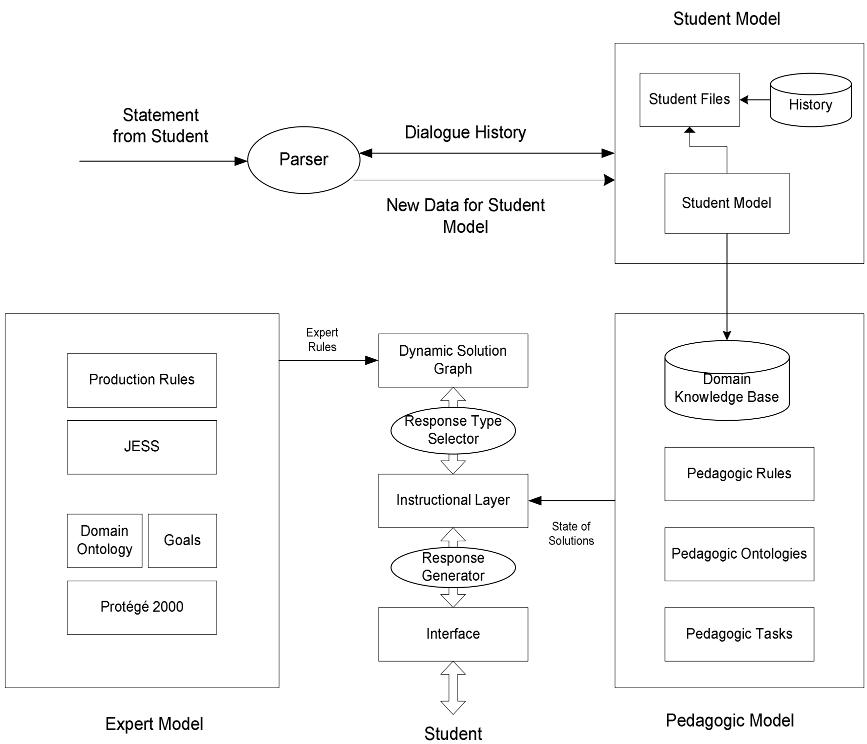


Fig. 1. Proposed Architecture of Merlin version 3

The student model, used as a reference, will maintain each individual's learning status. The student modeling shell for the system architecture is divided into a few components such as Personal data, Performance, Teaching History, Preferences, Navigation and Learning Styles. The student model for each component is an instantiation from the student model design pattern. Parameters in the student model can be edited without affecting other unrelated components in the architecture. The Behavior Analyzer Agent will classify user problems (received from the Detection Agent) into errors and pass the information to the pedagogical module. Different agents with their respective pedagogical abilities will solve the user problems with the help from a student model.

The pedagogic model consists of Domain Knowledge Base, pedagogic rules, pedagogic ontology and pedagogic tasks. The Domain Knowledge Base Manager will help in coordinating the users to use and update all kinds of knowledge stored in the

Knowledge Base. The pedagogic model will pass the state of solutions to the Instructional Layer, which invokes the Response Generator whenever feedback is needed. Consequently, the Response Generator will initiate a response and control will be passed to the Interface Agent, which plays the role of displaying the interface to the students.

The Expertise Layer will endow the agent with pedagogical intelligence. The production rules will be generated using JESS whereas Protégé 2000 is used to generate domain ontology and goal. A Learning Tutor Agent, which has both a tutoring engine and an inference engine, will be included in this model as well.

The Expert Model will pass expert rules to the dynamic solution graph, which will invoke the Response Type Selector to select the response to be passed to the Instructional Layer.

2.2 Research Questions

The scope of our investigation aims to design a framework for intelligent tutoring in collaborative learning systems using SOA. Specific research questions being investigated in this paper are:

1. How can we create a better personalized collaborative learning environment whereby users can interact with the system according to their needs and levels?
2. How can we determine the teaching strategy, which provides suitable assistance according to different user groups?
3. E-learning, intelligent tutoring systems, learning services and learning companion services have been implemented in a traditional manner whereby there is little possibility of reuse due to the tight coupling of components and lack of reuse for learning activities. So how can we incorporate learning companion agents in a systematic manner in order to achieve generalization and modularization of system services so that everyone can reuse, mix and match services suitable to their needs?

Outline of the paper is as follows: - Student models are identified. Teaching strategies are determined for different groups of users. Based on the matching results of new student profile and existing students' models, teaching strategy, content and services are recommended. The services include learning services and learning companion services. We designed an approach based on a proven Service-Oriented Analysis and Design (SOAD) methodology proposed by [8] to implement the services in improving generalization and modularization of components, pedagogy and learning activities.

3 Related Works

Based on the learning styles and capabilities, knowledge states and performance, learners can be grouped into categories, which ease the effort in locating adapted teaching strategy for them. In this paper, students are categorized into advance, intermediate and novice. An example of how the categorization can be done is shown in the simulation section (4.2).

3.1 Teaching Strategies

Instructional strategies are based on desired learning outcomes and learners' prior knowledge, experience and interests [9], [10] and [11]. Direct instruction may be more efficient and appropriate than constructivist approaches when teaching basic procedures (e.g. simple algorithms). In contrast, when the primary learning objective is to have students critically analyze, interpret and apply an ill-defined body of research, constructivist approaches may be more appropriate than teacher-directed methods. Students differ in terms of learning requirements and preferred learning methods [9], [10] and [12]. To satisfy as many students as possible, different approaches should be utilized. Like students, instructors have their own strengths, weakness and beliefs and they should apply what is best suited to their teaching and management style. Essential teaching and learning strategies include: collaborative learning, cognitive apprenticeship, discovery learning, worked example, mastery learning, problem-based learning, tutorial approach, goal-based scenarios, case-based teaching, guided design, anchored instruction, project-based models.

3.2 Learning Companion Agents

Broader in scope than ITS, the learning companion covers both aspects of tutoring and learning together with users. The learning companion helps to stimulate the student's learning through the process of collaboration and competition. It assists in guiding users to achieve their learning goals through tutoring and coaching. The learning companion agent plays different roles in different learning systems. Several literatures have proposed different variations of possible learning companion agents, which have different characteristics for example *collaborator, troublemaker, tutor, clone, tutee*, and etc.

Learning companion as a collaborator/competitor

Learning companion systems play different instructional roles. [13] have proposed that a learning companion system should involve three agents namely the human student, the computer-learning companion and the computer- teacher (as shown in Figure 2).

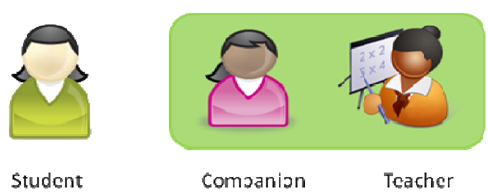


Fig. 2. Learning companion model

The role of the computer teacher is to offer examples, guidance and comments to both the student and the learning companion. The goal of the learning companion is to stimulate the student's learning through the process of collaboration and competition. [14] have mentioned how capable the collaborating companion agent should be in order to be a meaningful acquaintance for the human learner. In this approach, the question of relevant and sufficient interaction vocabulary is of utmost importance. Different types of learning companions with different levels of knowledge and expertise are applied.

Learning Companion as a tutor/tutee

[15] have presented a novel simulation approach, named General Companion Modeling (GCM), to implement learning companions in a general problem-solving domain. DwestAgent, a learning companion system, is implemented using the GCM approach to simulate 1) domain competencies, 2) learning competencies, 3) behaviors as a peer tutor, and 4) behaviors as a peer tutee of a learning companion. As shown in Figure 3, the GCM architecture consists of four main components: behavior module for simulating the learning companion's behavior, domain module to simulate the domain competency of a learning companion, user model to store the user's status observed by a system, and learning companion pattern, which stores the learning companion's characteristic data.

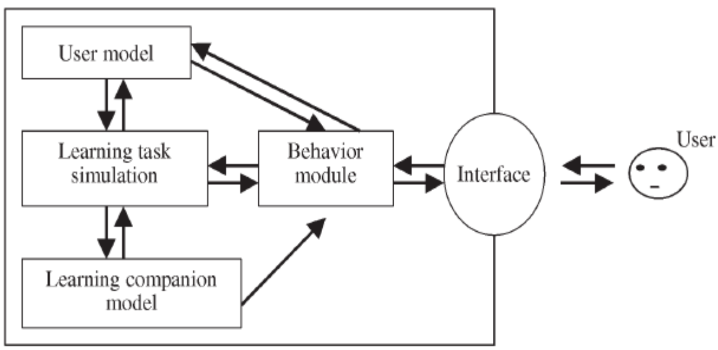


Fig. 3. GCM architecture

3.3 Service-Oriented Architecture (SOA)

There is an increased tendency to automate e-business contacts. However, the barrier is a lack of interoperability between different software systems used by business partners. A need for customization arises, where service users are able to fit the service into their individual requirements and expectations. In the context of e-learning, SOA plays an important role in enhancing interoperability among heterogeneous hardware and software platforms, integration of new and legacy systems and flexibility in updating software.

[16] has shown that SOA facilitates the reutilization of successful collaborative learning experiences and makes it possible for the collaborative learning participants to easily adapt and integrate their current best practices and existing well-known learning tools into new learning goals. The benefits of modularizing the learning components in collaborative learning environments include easier integration and increased interoperability. SOA can be realized using Web Services whereby the core structure is formed by a set of widely adopted protocols and standards such as XML, SOAP, WSDL and UDDI. These protocols allow a service to be platform- and language- independent, dynamically located and invoked and interoperable over different organizational environments.

[16] has identified the core services required to support collaborative learning applications and categorized them into common and application services. Common services provide lower-level functionality, which is not education-specific. However, educational-domain services and users depend on these common services. On the contrary, application services are educationally domain dependent and provide the functionality required by agents. Common services in collaborative learning systems include Authentication, Authorization, Messaging, Logging, Search and so on. On the other side, application services include Sequencing, Content Management, Assessment, Communication, Coordination, Grading and etc. The application services' key requirement is to expose their functionality for reuse by any number of agents or other application services.

4 Proposed Solution

The overall architecture of incorporating SOA in implementing traditional models of the ITS and learning companion is shown in Figure 4. The Domain Model includes the domain knowledge and is presented to the students. The Student Model represents the student knowledge about the domain. This model logs the history of the students' navigation on the learning systems. It also includes the Learning Companion Services chosen based on the students' profile. The Pedagogical Model decides the level of knowledge and types of the Learning Services/ Learning Companion Services that will be presented to the student, based on their performance. It keeps tracks of the students' performance and decides the types of Learning Companion Services for the students.

4.1 SOA Layers

By applying SOAD methodology by [8], the design of the main architecture for implementing learning/learning companion services using SOA is shown in Figure 4. There are four layers in the proposed architecture namely:

1. Application layer - presents the service invocation results to the users via client program/designed user interface.
2. Business Process layer - decides the workflow of service invocation from the Learning Services Layer or Learning Companion Services Layer based on the learning tasks given and also the student profile. The business processes

(BP) within the models (competitor, collaborator, tutor, etc) in the Learning Companion Services Layer are to be identified. There exists variability in the BP for each learning companion service personalized to each user. In an effort to personalize the service, the BP in each learning companion service is connected to the knowledge base.

3. Learning Services Layer - consists of all types of common learning services in collaborative systems such as concept mapping, document sharing, discussion forum, assessment, practices and etc.
4. Learning Companion Services Layer - includes the services of teaching strategies in the aspect of collaborator, competitor, tutor, and others.

Each service in the Learning Services Layer and Learning Companion Services Layer is modular and independent.

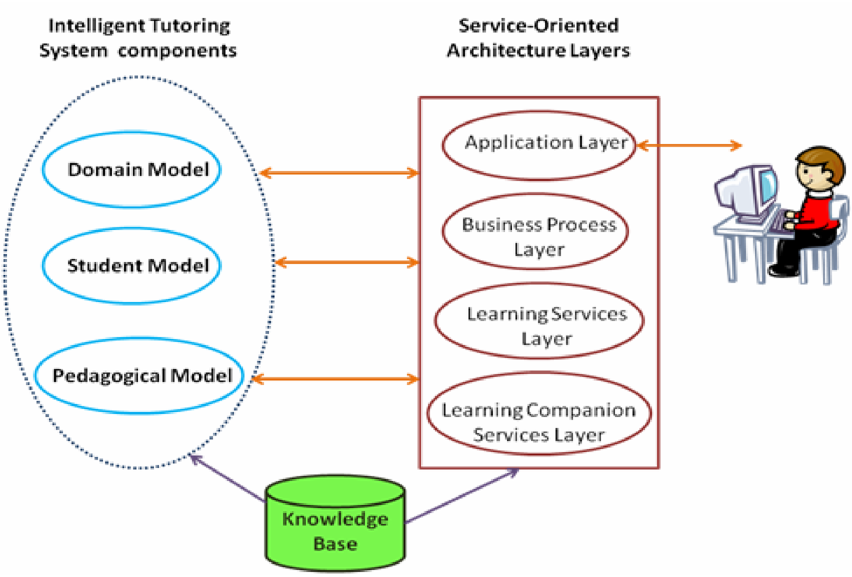


Fig. 4. Overall Architecture

4.2 Simulation

A simulation is performed to show the process of invoking personalized web services in distributed environment based on the students' level. The study was conducted during the undertaking of the course –'TNA2314 Numerical Analysis'. Our methodology is as follows:

1. Data is gathered from 80 scripts on how the students solve one particular Mathematics problem (Differentiation problem as shown in Table 1). The student data forms the training set as shown in Table 2.

2. Identify the student clusters (*novice, intermediate, advanced*) based on four rules determined by experts:
   - Number of steps generated to solve the problem
   - Number of rules used to solve the problem
   - Number of correct sequences of the solution
   - Solution result
3. Determine teaching strategies for different sets of students so the new student profile will invoke the suitable teaching strategy, which leads to invoke the content and then invoke the tutoring services.
4. Match a new student profile with the existing student models to identify which cluster the new student belongs in.
5. Based on the matching results, recommend the teaching strategy, content and services, which are relevant for that cluster to the student.

Table 1. An Example of Applying Basic Rules of Differentiation

Problem: $\frac{d}{dx}(\sin^2(x) + 7x - 3)$	
Working Steps	**Rules**
$= \frac{d}{dx}(-3) + \frac{d}{dx}(7x) + \frac{d}{dx}(\sin^2(x))$	The derivative of a sum is the sum of the derivatives
$= \frac{d}{dx}(7x) + \frac{d}{dx}(\sin^2(x))$	The derivative of the constant -3 is 0
$= 7\frac{d}{dx}(x) + \frac{d}{dx}(\sin^2(x))$	The derivative of a constant times a function is the constant times the derivative of the function
$= \frac{d}{dx}(\sin^2(x)) + 7$	The derivative of x^n *is* nx^{n-1}
$= 2\sin(x)\frac{d}{dx}(\sin(x)) + 7$	Use the chain rule $\frac{du^n}{dx} = n\,u^{n-1}\frac{du}{dx}$, *where* $u = \sin(x)$ *and* $n = 2$
$= 2\cos(x)\sin(x) + 7$	The derivative of $\sin(x)$ *is* $\cos(x)$
$= \sin(2x) + 7$	Simplify

We have sought advice from three Mathematics experts in defining the calculation of student models. Three experts have the average of 7 years in teaching Mathematics subjects. The weightage, ω carried for each rule and the thresholds, λ for identifying the students' level are determined by the experts. The weightage distributions are as follows:-

1. Number of steps generated to solve the problem (with the weightage of 5%)
2. Number of rules used to solve the problem (with the weightage of 40%)
3. Number of correct sequences of the solution (with the weightage of 45%)
4. Solution result (with the weightage of 10%)

The calculation of the ***Total Scores*** for each student:-

*Score for rule 1 * weightage* (ω) + *score for rule 2 * weightage* (ω) + *score for rule 3 * weightage* (ω) + *score for rule 4 * weightage* (ω)

The thresholds λ were determined based on this particular group of students only. The values were derived from all the scores obtained from the students which revolving from 1-6 and majority of the students scored between 2-4 which deriving the thresholds values for *intermediate* level students. The cut off *threshold,* λ for identifying the *novice, intermediate and advanced* students based on the ***Total Scores***:-

Novice, $\lambda < 2$
Intermediate, $2 \leq \lambda \leq 4$
Advanced, $\lambda > 4$

Table 2. Data Set

Students / Rules	1	2	3	4	5	6	7	8	9	10	n	80
1. Number of steps generated	8	5	5	7	5	5	4	6	5	4	..	4
2. Number of rules used	2	3	3	6	1	2	4	4	3	4	..	4
3. Number of correct sequences	2	3	3	6	1	0	3	4	3	3	..	3
4. Solution result	Wrong	True	True	True	Wrong	Wrong	True	True	True	True	…	True

Based on the results obtained as shown in Table 3, we can conclude that 13 students are grouped in the advanced category, 51 students are grouped as intermediate and 16 as novice.

Table 3. Data Set of the total scores

Rules \ Students	1	2	3	4	5	6	7	8	9	10	n	80
1. Number of steps generated (ω=0.05)	0.4	0.25	0.25	0.28	0.25	0.25	0.2	0.3	0.25	0.2	…	0.2
2. Number of rules used (ω=0.4)	0.8	1.2	1.2	2.4	0.4	0.8	1.6	1.6	1.2	1.6	…	1.6
3. Number of correct sequences (ω=0.45)	0.9	1.35	1.35	2.7	0.45	0	1.35	1.8	1.35	1.35	…	1.8
4. Solution result (ω=0.1)	0	0.1	0.1	0.1	0	0	0.1	0.1	0.1	0.1	…	0.1
Total Scores	2.1	2.9	2.9	5.48	1.1	1.05	3.25	3.8	2.9	3.25	…	3.7

Upon knowing the student level, when there's new student model, we perform classification on it to group it into its associated student cluster (*novice, intermediate, advanced*). Consequently, depending on the problems faced in solving the Differentiation task, personalized teaching strategy, which is from Learning Services or Learning Companion Services Layer is invoked (as shown in Table 4).

Table 4. Some of the Rules and invoked teaching strategies

Rules **Section 1. Solving differentiation**	**Teaching Strategies**
Rule 1 ***IF** task is 'solving differentiation'* *THEN ask problem* **Section 1.1 Wrong Solution** ***IF** task is 'solving differentiation'* *AND problem is 'wrong solution'* *AND student type is 'novice'* *THEN ask 'teaching strategy **A**'* ***IF** task is 'solving differentiation'*	**Web Service A** provides working examples with solutions for solving the differentiation problem whereby the steps are being explained in details. It serves as a tutor. **Web Service B** provides tutorial approach whereby the working examples are given without explanations. Case-based teaching is practiced. Students develop skills in analytical thinking and reflective judgment by reading and discussing the real examples.

Table 4. *(continued)*

AND problem is 'wrong solution' *AND student type is 'intermediate'* *THEN ask 'teaching strategy **B**'* ***IF** task is 'solving differentiation'* *AND problem is 'wrong solution'* *AND student type is 'advanced'* *THEN ask 'teaching strategy **C**'*	**Web Service C** acts as a tutee to encourage the students to explore the problem by themselves while teaching the tutee. Discovery learning is practiced.
Section 1.2 Usage of Rules ***IF** task is 'solving differentiation'* *AND problem is 'rules are needed'* *AND student type is 'novice'* *THEN ask 'teaching strategy **D**'* ***IF** task is 'solving differentiation'* *AND problem is 'rules are needed'* *AND student type is 'intermediate* *THEN ask 'teaching strategy **E**'* ***IF** task is 'solving differentiation'* *AND problem is 'rules are needed'* *AND student type is 'advanced'* *THEN ask 'teaching strategy **F**'*	**Web Service D** provides all the rules needed in details whereby some working examples, which utilizes the rules are provided. It serves as a tutor. **Web Service E** provides all the rules needed. No working examples are shown. **Web Service F** acts as the collaborator in helping to identify the needed rules. The collaboration increases the students' interactivity thus improving the problem solving capability.
Section 1.3 Wrong sequences ***IF** task is 'solving differentiation'* *AND problem is 'wrong sequences'* *AND student type is 'novice'* *THEN ask 'teaching strategy **G**'* ***IF** task is 'solving differentiation'* *AND problem is 'wrong sequences'* *AND student type is 'intermediate'* *THEN ask 'teaching strategy **H**'* ***IF** task is 'solving differentiation'* *AND problem is 'wrong sequences'* *AND student type is 'advanced'* *THEN ask 'teaching strategy **I**'*	**Web Service G** identifies all the wrong sequences and recommends the correct sequences for the students. Explanations are provided. **Web Service H** practices problem-based learning whereby it provides carefully-designed problems to challenge the students to use problem solving techniques, self-directed learning strategies, and team participation skills in identifying the correct sequence in the working steps. **Web Service I** acts as a competitor whereby students need to compete with the service to identify the correct sequences. This allows students to discover the mistakes by themselves in a competitive environment.

5 Conclusion

Collaborative learning practices are constantly adaptive and personalized to a target user group. Due to distributed resources everywhere, SOA plays an important role to enable reuse of components, pedagogy and learning activities in this heterogeneous environment. We have developed the framework for a better personalized learning environment whereby users can interact with the system according to their needs and levels. Teaching strategies, which are implemented as web services provide suitable assistance according to different users group. Users may invoke the web services according to their learning needs which highly promote the reusability of the teaching strategies. For our future works and exploration, issues such as services' variability and adaptability are to be considered when designing and executing the learning service process. Quality of web services' attributes should be included too in the learning service discovery and selection process. We also aim to improve the artificial intelligent techniques applied in our learning system to facilitate better quality user learning experiences.

References

1. Waterhouse, S. : The power of e-learning the past, the present, and the future (2003), `http://ritim.cba.uri.edu/wp2003/pdf_format/Wiley-Encycl-Internet-Diffusion-v12.pdf`
2. Rosenberg, M.J.: E-learning: Strategies for delivering knowledge in the digital age. McGraw-Hill, New York (2001)
3. Lee, C.S.: Problem-solving in a constructivist environment. Educational Technology and Society 2(4), 137–145 (1999)
4. Piaget, J.: The Development of Thought: Equilibration of Cognitive Structures. Viking, New York (1977)
5. Joseph, P., Mutter, S.A.: Intelligent Tutoring Systems: Lessons Learned, vol. 2(4), pp. 137–145. Lawrence Erlbaum Associates, Mahwah (1988)
6. Chua, F.F., Lee, C.S.: The effects of learning goals and modeling on virtual collaboration: a case study on Merlin agent-assisted collaborative concept map. International Journal of Web-based Communities 3(2), 229–248 (2007), Inderscience [INSPEC] ISSN: 1741-8216
7. Chua, F.F., Lee, C.S.: Designing an Intelligent Coach for a Collaborative Concept-Mapping Learning Environment. In: Proceedings of the 7th International Conference on Web Based Learning, Zhejiang, China, August 20-22. LNCS (2008)
8. Chang, S.H., Chua, F.F., Kim, S.D.: An Approach for Verification in Service-Oriented Computing. In: IEEE Congress on Services-Part I, July 8-11, pp. 575–582 (2008)
9. Seels, B.: Instructional design movement in educational technology. Educational Technology 29(5), 11–15 (1989)
10. Seels, B., Glasgow, Z.: Making instructional design decisions, 2nd edn. Prentice-Hall, New Jersey (1998)
11. Rogers, P.L.: Teacher-designers: How teachers use instructional design in real classrooms. In: Designing Instruction for Technology-Enhanced Learning. Idea Group Publishing, USA (2002)
12. Schweizer, H.: Designing and teaching an on-line course: Spinning your web classroom. Allyn & Bacon, USA (1999)

13. Chan, T.W., Baskin, A.B.: Learning companion systems. In: Frasson, C., Gauthier, G. (eds.) Intelligent Tutoring Systems: At the Crossroads of Artificial Intelli-gence and Education, ch. 1, Ablex Publishing Corporation, New Jersey (1990)
14. Hietala, P., Niemirepo, T.: Collaboration with software agents: what if the learning companion agent makes errors? In: du Boulay, B., Mizoguchi, R. (eds.) Proceedings of AIED 1997 World Conference on Artificial Intelligence in Education, Kobe, Japan, pp. 159–166. IOS Press, Amsterdam (1997)
15. Chou, C.-Y., Chan, T.-W., Lin, C.-J.: An approach of implementing general learning companions for problem solving. IEEE Transactions on Knowledge and Data Engineering 14(6), 1376–1386 (2002)
16. Santi, C.: On the Advantages of Using Web & Grid Services for the Development of Collaborative Learning Management Systems. In: The Proceedings of the First Interna-tional Conference on Complex, Intelligent and Software Intensive Systems, pp. 263–270 (2007)

Probability Modelling of Accesses to the Course Activities in the Web-Based Educational System

Michal Munk, Martin Drlik, and Marta Vrábelová

Constantine the Philosopher University in Nitra, Tr. A. Hlinku 1,
949 74, Nitra, Slovakia
{mmunk,mdrlik,mvrabelova}@ukf.sk

Abstract. The aim of the paper is the probability modelling of accesses to the categories of activities of e-learning course in learning management system. We are concerned with the access probabilities to individual activities of e-learning course content depending on the part of the week (workweek and weekend). The probabilities are estimated through multinomial logit model. We pay attention to data preparation issues. We describe used model in more detail and deal with parameter estimations. Finally, we figure that the multinomial logit model finds its application mainly in the process of restructuring the existing e-learning courses. We discuss about its possible contribution to the improvement of the learning management as well as in the personalization of the course content and structure.

Keywords: web log mining, user behaviour, learning management, e-learning course content, probability modelling, multinomial logit model.

1 Introduction

One of the first tasks of web mining is the analysis of behaviour of website visitors. Based on this analysis, in educational contexts, it is possible to personalize e-learning courses, adapt educational hypermedia, discover potential browsing problems, automatically identify learning groups in exploratory learning environments or predict student's performance. The analysis of the unique types of data coming from educational systems can help find the most effective structure of e-learning courses, optimize the learning content, recommend the most suitable learning path based on student's behaviour, or provide more personalized environment.

The aim of the paper is the probability modelling of accesses to the categories of activities in e-learning course of LMS portal at the university. For this purpose, data about accesses to the activities of the e-learning course were collected. We are concerned with the access probabilities to individual activities of e-learning course content depending on the part of the week (workweek and weekend). The probabilities are estimated through multinomial logit model [1, 2] for students connected from inside and outside of the university network separately.

The multiple logistic regression model that we will use in this paper is fully described in the book [3]. This model is a special type of the general linear model.

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 485–499, 2011.

The basics of this theory is obtained in the book [4] or [2]. We can find its applications mostly in econometrics, genetics and natural language processing. Its usage in the same research area as we describe in our paper is relatively infrequent. If the multiple logistic regression model is used then it is used mainly for choice prediction [5, 6].

In general, discovering association and sequence rules, segmentation (Cluster analysis, methods based on analogy, etc.), and classification (decision rules, decision trees, Bayes classification, etc.) [7] are the most applied methods in the web log mining. Out of the statistic methods we apply e.g. analysis of crosstabulations relative to the categorical character of data [8]. Data reading into a data cube is another often used method to carry out OLAP operations for data summarization [9], but some tools combine OLAP and knowledge discovery [10]. Automatically saved data in log files are used as a source of the data, which are, from the point of knowledge discovery, represented by time data. Most methods applied to such data are used to perform segmentation of web visitors, extraction of behaviour patterns of web visitors, and finding associations among visited websites with the aim to personalise or optimise (restructure) websites according to the way they are browsed. The same applies to the web-based educational system [11-14]. But neither of the above mentioned methods models the behaviour of the user depending on time. Exactly for this purpose we used multinomial logit model for modelling the probability of accesses depending on time.

The rest of the paper is structured as follows. The next chapter deals with the data pre-processing tasks. We describe used multinomial logit model in third chapter. The fourth chapter describes the determination of the model and parameter estimations. We deal with its evaluation in fifth chapter. Finally, we discuss about results and figure that the multinomial regression model finds its application in various stages of the e-learning course's development life cycle, as well as in the personalization of the course content and learning management.

2 Data Preparation

The information available on the web is heterogeneous and unstructured. The goal of data preparation is to transform the raw data stored in logs into a set of user profiles [15]. Contemporary learning management systems (LMS) store information about their users not in server log file but mainly in a relational database. LMS manage all their services through a relational database. We can find there large log data of students' activities and, usually, LMS have built-in student monitoring features so they can record all kinds of student activities [16]. Compared to other data mining applications [17], a relational database provides an integrated source of data saving preprocessing effort.

In the e-learning context, unlike other web based domains, user identification is a straightforward problem as in most cases learners must login using their unique ID [18]. We used log file created from database containing records from the e-learning course with 180 participants. The e-learning course has been created in LMS Moodle. The log file has been dumped from the tables mdl_log and mdl_log_display.

Records have been cleaned from irrelevant items. First of all, we have removed entries about users with the role other then student. After performing this task, 70 553 entries were accepted to be used in the next task.

A user may have a single (or multiple) session(s) during a period of time. We have found many approaches to session identification [9, 19-23] and [24]. Against this background we have used reactive time-oriented heuristic method to define the users' sessions and we have considered not only user's ID, but also the IP address of a computer used by the user and 15-minute timeout.

We have taken decision to identify users' sessions not by the reason of analysis requirements but for the purpose of correct path reconstruction. Path reconstruction means the analysis of the backward path, or reconstruction of activities of a web visitor. We found and analyzed several approaches mentioned in literature [23, 24]. Finally, we have chosen the same approach as in our previous paper [25]. After performing this task, the count of records increases to 75 372, which means 7% increase of records.

The calculation of some derived variables is the last step of data-preprocessing stage. The original logfile contains only the variable ACCESS_TIME. We had to calculate another two variables, WEEK (weekend and workweek) and TIME (hours in day).

3 Model Description

First we describe the variables which we need to include in the model. The investigated categorical dependent variable is a variable ACTIVITY with categories: *assignment, book, blog, course, data* (i.e. files, home works and projects uploaded by students), *feedback, forum, glossary, quiz, resource* (i.e. information sources available in the course), *upload* and *user* (student's profile, his/her grades and information about other students).

The variable TIME with values 0-23 is non-dependent variable. We use variable WEEK with categories *weekend* (0) and *workweek* (1) and the variable ACCESS_PLACE with *in* (0) and *out* (1) as dummy variables. Figures 6a and 6b show the empirical (and fitted) logits of activities during the workweek for internal accesses and for external accesses (using the last activity *user* as the reference category) plotted against the time. From the figures it seems to follow that the logits are quadratic functions of time. Similar results were obtained for the weekend. We will therefore include the variable square time TIME^2 in the model.

The data describes individual accesses of students into e-learning course during the winter term that was created in LMS Moodle. Denote by π_{ij} the probability of selecting the activity j by a student in the hour i, where $j = 1, 2, ..., J$.

Since, $\sum_{j=1}^{J} \pi_{ij} = 1$ there are J - 1 parameters [26]. Let Y_{ij} be the number of selecting the activity j in the hour i, with observed value y_{ij}.

Then $\sum_{j=1}^{J} \pi_{ij} = 1$ is the number of selecting in the hour i. The probability distribution of the vector $Y_i = (Y_{i1}, Y_{i2}, \dots Y_{iJ})^T$, in the case n_i is given, is multinomial,

$$f_i\left(y_{i1}, y_{i2}, \ldots, y_{iJ}\right) = P\left[Y_{i1} = y_{i1}, Y_{i2} = y_{i2}, \ldots, Y_{iJ} = y_{iJ}\right] = \frac{n_i\,!}{y_{i1}\,!\,y_{i2}\,!\ldots y_{iJ}\,!}\pi_{i1}^{y_{i1}}\pi_{i2}^{y_{i2}}\ldots\pi_{iJ}^{y_{iJ}} \tag{1}$$

Taking logs we find that

$$\log f_i\left(y_{i1}, y_{i2}, \ldots, y_{iJ}\right) = \sum_{j=1}^{J} y_{i1}\log\ \pi_{ij} + \log\left(\frac{n_i\,!}{y_{i1}\,!\,y_{i2}\,!\ldots y_{iJ}\,!}\right)$$

Since $\sum_{j=1}^{J}\pi_{ij} = 1$, we put $\pi_{iJ} = 1 - \sum_{j=1}^{J-1}\pi_{ij}$ and we get

$$\log f_i\left(y_{i1}, y_{i2}, \ldots, y_{iJ}\right) = \sum_{j=1}^{J-1} y_{ij}\log\ \pi_{ij} + \left(n_i - \sum_{i=1}^{J-1} y_{ij}\right)\log\left(1 - \sum_{j=1}^{J-1}\pi_{ij}\right) + \log\left(\frac{n_i\,!}{y_{i1}\,!\,y_{i2}\,!\ldots y_{iJ}\,!}\right)$$
$$= \sum_{j=1}^{J-1} y_{ij}\log\frac{\pi_{ij}}{1 - \sum_{j=1}^{J-1}\pi_{ij}} + n_i\log\left(1 - \sum_{j=1}^{J-1}\pi_{ij}\right) + \log\left(\frac{n_i\,!}{y_{i1}\,!\,y_{i2}\,!\ldots y_{iJ}\,!}\right).$$

Put

$$\eta_{ij} = \log\frac{\pi_{ij}}{\pi_{iJ}} \text{ and } \boldsymbol{\eta}_i = \left(\eta_{i1}, \eta_{i2}, \ldots, \eta_{iJ}\right)^T,\ \ \mathbf{y}_i = \left(y_{i1}, y_{i2}, \ldots, y_{iJ}\right)^T.$$

Then the probability distribution function has the general exponential form

$$f_i\left(\mathbf{y}_i\right) = C\left(\boldsymbol{\eta}_i\right)\exp\left(\sum_{j=1}^{J-1} Q_j\left(\boldsymbol{\eta}_i\right)T_j\left(\mathbf{y}_i\right)\right)u\left(\mathbf{y}_i\right), \text{ where}$$

$$C\left(\eta_i\right) = \pi_{iJ}^{n_i},\ \ Q_j\left(\boldsymbol{\eta}_i\right) = \eta_{ij},\ \ T_j\left(\mathbf{y}_i\right) = y_{ij},\ \ \mathrm{u}\left(\mathbf{y}_i\right) = \frac{n_i\,!}{y_{i1}\,!\,y_{i2}\,!\ldots y_{iJ}\,!}.$$

Hence, we can use the generalized linear model with link function logit to estimate the probabilities π_{ij} of selecting activity j with respect to the hour i . We assume the following model

$$\eta_{ij} = \log\frac{\pi_{ij}}{\pi_{iJ}} = \mathbf{x}_i^T\boldsymbol{\beta}_j,\quad j = 1, 2, \ldots, J-1,\ i \in \left\{0, 1, \ldots, 23\right\}, \tag{2}$$

where π_{iJ} is the probability of last (reference) activity, $\mathbf{x}_i^T$ is a line vector, $\boldsymbol{\beta}_j = \left(\beta_{j1}, \beta_{j2}, \ldots, \beta_{jk}\right)^T$ is a vector of regression coefficients for $j = 1, 2, \ldots, J-1$. There are J - 1 equations which contrast each of activities $1, 2, \ldots, J-1$ with the category J. The probabilities π_{ij} we obtain from formulas

$$\pi_{iJ} = \frac{1}{1 + \sum_{k=1}^{J-1} e^{\eta_{ik}}},\ \ \pi_{ij} = e^{\eta_{ij}}\pi_{iJ},\ \ j = 1, 2, \ldots, J-1. \tag{3}$$

Maximum likelihood estimation of the parameters of the model (2) proceeds by maximization of the multinomial log-likelihood (without the constants)

$$\log L(\boldsymbol{\alpha},\boldsymbol{\beta},\mathbf{y}) = \sum_{i=0}^{23}\sum_{j=1}^{J} y_{i1} \log \pi_{ij}$$

with the probabilities π_{ij} viewed as functions of $\boldsymbol{\beta}_j$ by (3) and (2). The estimation of the parameters can be done using an iteratively re-weighted least squares method as the Newton-Raphson technique or Fisher scoring (see Rodríguez [2], Appendix A). The starting values of estimates $\boldsymbol{\beta}_{j0}$ are computed from empirical logits

$$\eta_{ij0} = \log \frac{n_{ij}}{n_{iJ}} = \mathbf{x}_i^T \boldsymbol{\beta}_j, \quad j = 1,2,\ldots,J-1,\ i \in \{0,1,\ldots,23\}$$

by linear regression.

The maximum likelihood estimator $\widehat{\boldsymbol{\beta}}_j$ has approximately in large samples a multivariate normal distribution with mean equals to the true parameter value and variance-covariance matrix given by the inverse of the information matrix. The hypothesis H_0: $\boldsymbol{\beta}_i = \mathbf{0}$ can be tested by the Wald test.

When expected counts $\widehat{\mu}_{ij} = n_i \widehat{\pi}_{ij}$ are large enough (that is no more than 20% of the μ_{ij}'s are below 5 and none are below 1) for comparing current model to a saturated model that estimates the probabilities independently for $i = 0,1,\ldots,23$ we can use the deviance

$$G^2 = 2\sum_{i=0}^{23}\sum_{j=1}^{J} y_{ij} \log \frac{y_{ij}}{\mu_{ij}}.$$

The saturated model has $24(J - 1)$ free parameters and the current model has $k(J - 1)$, the degrees of freedom are $df = (24 - k)(J - 1)$. The statistics G^2 has approximately $\chi^2(df)$ distribution. The Pearson statistics is

$$\chi^2 = \sum_{i=0}^{23}\sum_{j=1}^{J} r_{ij}^2, \text{ where } r_{ij} = \frac{y_{ij} - \widehat{\mu}_{ij}}{\sqrt{\widehat{\mu}_{ij}}}$$ is the Pearson residual, has $\chi^2(df)$ distribution,

too.

4 Model Determination and Parameter Estimation

In the following part the accesses to the course will be described through association rules. The association rules analysis represents a non-sequential approach to the analyzed data. Not sequences, but transactions will be analyzed, i.e. the analysis will not include time variable. In this case, transaction is a set of course activities accessed by a student during one session.

The web graph (Fig. 1) is a visualization of the determined association rules; in particular the size of the knot shows support of the element, thickness of line shows support of the rule, and brightness of line shows lift of the rule.

From the preceding graph (Fig. 1), which transparently describes selected associations, may be determined that the most frequently accessed course activities include *course* (*support* = 77 %), *resource, data, assignment, quiz* (*support* > 15 %), as well as combinations of the activity *course* with the other activities (*support* > 10 %).

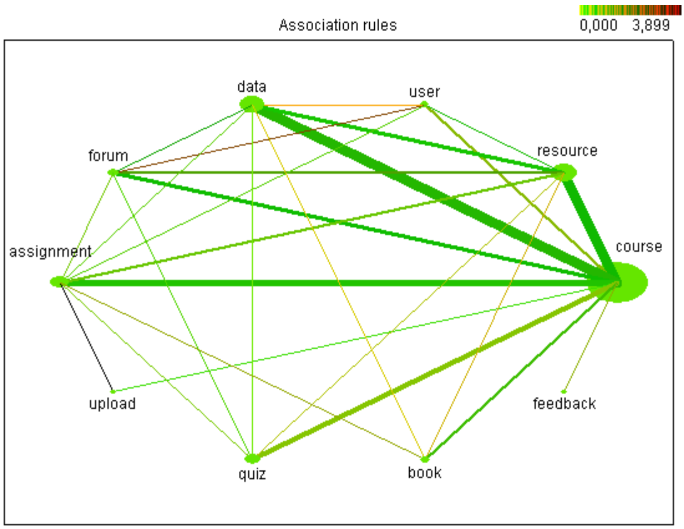


Fig. 1. Web graph – visualization of the found rules

It may also be seen (Fig. 1) that the *assignment* and *upload* occur more frequently together in the sets of the course accessed activities than separately (*lift* = 3.9). The same holds true for the activities *forum* and *user* (*lift* = 2,8). In these cases the highest degree of interest was reached for the lift, determining how more often the accessed activities occur together than if they were statistically independent. If the lift happens to be more than one, the pairs of activities occur more often together than separately in the sets of course activities accessed by students in individual sessions.

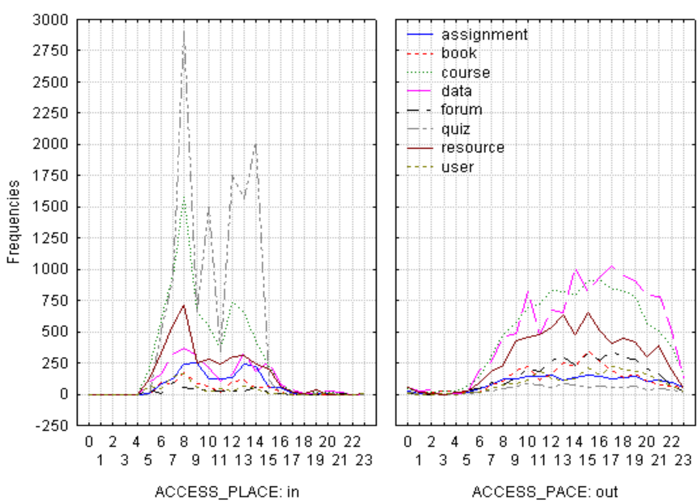


Fig. 2. Interaction plot – ACTIVITY x ACCESS_PLACE x TIME

On the contrary, the least accessed course activities (Fig. 1) include the *upload* (*support* = 3 %), *feedback* (*support* = 2.5 %), and the activities *glossary* and *blog* which did not reach even the minimum value of support (*min support* = 1 %), as well as combinations *upload, assignment* and *feedback, course* (*support* = 2 %), and the combination *upload, course* (*support* = 1 %).

Based on the results of the course access analysis, the least accessed activities, i.e. the *upload, feedback, glossary*, and *blog*, will be excluded from further analyses.

We have found out, with respect to contingence tables (Fig. 2), that the significant numbers of accesses for variables ACTIVITY, TIME and ACCESS_PLACE are:

- for internal accesses (*in*) in the hours 5-19, to the activities *assignment, book, course, data, forum, quiz, resource* and *user,*
- for external accesses (*out*) in the hours 0-23, to the activities *assignment, book, course, data, forum, quiz, resource* and *user.*

Table 1. The numbers of accesses to activities

ACCESS	Total	ACTIVITY							
PLACE	count	assignment	book	course	data	forum	quiz	resource	user
in	29289	1678	867	7067	2620	412	12305	3668	672
out	42176	2166	2840	11452	12033	3529	959	6994	2203
Total	71465	3844	3707	18519	14653	3941	13264	10662	2875

Table 2. Tests of all effects; Likelihood type I test; Likelihood type III test

	Model	df	Wald Stat.	p	Log-Likelihood	Chi-Square	p	Log-Likelihood	Chi-Square	p
Intercept	in	7	383.9	0.000	-46807.5					
WEEK	in	7	182.9	0.000	-46587.7	439.5	0.000	-45949.3	407.9	0.000
TIME	in	7	843.7	0.000	-46151.6	872.2	0.000	-46215.9	941.0	0.000
TIME_Q	in	7	733.4	0.000	-45745.4	812.5	0.000	-46151.6	812.5	0.000
Intercept	out	7	512.0	0.000	-75568.4					
WEEK	out	7	301.7	0.000	-75415.5	305.9	0.000	-75314.8	315.1	0.000
TIME	out	7	176.2	0.000	-75248.8	333.3	0.000	-75247.5	180.5	0.000
TIME_Q	out	7	179.2	0.000	-75157.2	183.2	0.000	-75248.8	183.2	0.000

We have excluded non-significant accesses. The numbers of accesses to those activities are in Table 1.

We have used the following models. For internal accesses:

$$\eta_{ij} = \beta_{0j} + \beta_{1j} TIME_i + \beta_{2j} TIME_i^2 + \beta_{3j} WEEK_i, \quad j = 1, 2, \ldots, 7, \; i = 5, 6, \ldots, 19. \qquad (4)$$

For external accesses:

$$\eta_{ij} = \beta_{0j} + \beta_{1j} TIME_i + \beta_{2j} TIME_i^2 + \beta_{3j} WEEK_i, \quad j = 1, 2, \ldots, 7, \; i = 0, 1, \ldots, 23. \qquad (5)$$

Based on *Test of all effects and the Likelihood type I test* and *III tests* in the created logit models (Tab. 2), results present the week statistically significant sign, which is represented with dummy variable WEEK in the models. Hours, represented by variables TIME and their square $TIME^2$, were shown as statistically significant signs in both cases (for internal accesses and external accesses).

The parameters of the models were estimated in the *STATISTICA Generalized Linear/Nonlinear Models*. They are in Table 3. The significance of parameters was tested through *Wald test*; significant parameters are colored.

Table 3 shows that logits for individual activities are with external accesses significantly dependent on the hour of access as well as on its square and on the dummy variable WEEK, except for the *forum* activity, which does not depend on the hour of access, and, partially, the *book* activity which does not depend on its square.

On the contrary, in the case of internal accesses (from the university net), except for the *forum* activity, there is no significant dependence on the hour also for the activities *data* and *resource*. The values of these logits, in the case of external accesses, are significantly influenced by the variable WEEK which distinguishes *workweek* and *weekend*. On the other hand, with internal accesses the WEEK variable has a significant influence on the value of logits only for the activities *forum, quiz* and *resource.*

Table 3. The parameter estimations

	ACTIVITY	ACCESS_PLACE		ACTIVITY	ACCESS_PLACE	
		in	out		in	out
Intercept	assignment	-3.2016	2.4554	forum	1.0485	0.1831
TIME	assignment	0.8264	-0.3028	forum	0.1008	0.0510
TIME_Q	assignment	-0.0281	0.0092	forum	-0.0005	-0.0025
WEEK	assignment	-1.0433	-0.2969	forum	-2.4348	0.1764
Intercept	book	-0.5763	1.6841	quiz	-7.2987	0.7888
TIME	book	0.4277	-0.1149	quiz	1.3443	-0.2302
TIME_Q	book	-0.0130	0.0018	quiz	-0.0538	0.0059
WEEK	book	-1.9493	-0.2602	quiz	2.6482	0.4727
Intercept	course	1.0888	2.9225	resource	3.3272	2.4779
TIME	course	0.4691	-0.1378	resource	0.1246	-0.1173
TIME_Q	course	-0.0176	0.0036	resource	-0.0008	0.0023
WEEK	course	-1.4240	-0.1242	resource	-2.7311	-0.2114
Intercept	data	-0.3836	3.1041	user	-	-
TIME	data	0.0885	-0.1702	user	-	-
TIME_Q	data	0.0047	0.0058	user	-	-
WEEK	data	0.4353	-0.3908	user	-	-

Using the estimated parameters it is possible to enumerate the estimates of logits and the probabilities of selecting individual categories in any given day's hour. Then theoretical numbers of accesses to individual categories can be enumerated. This is proved by some illustrations. Graphs of the probabilities of selection of activities during the workweek (Fig. 3) and weekend (Fig. 4) by students who study at the university (in the classrooms/study halls) are in the figure (a) and the figure (b) obtains these probabilities for students who study outside the university (at work/home).

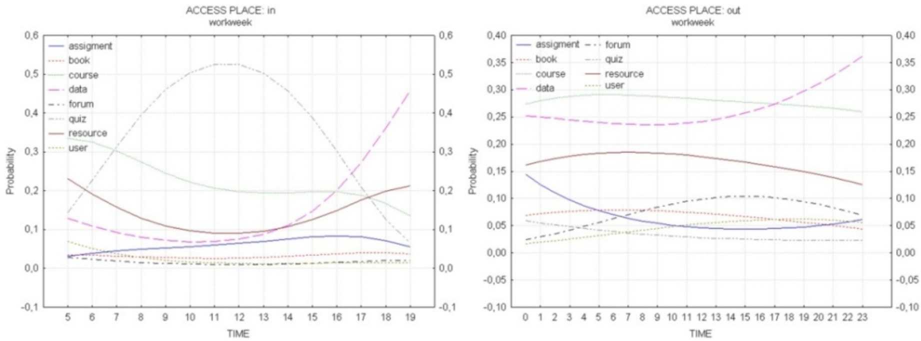


Fig. 3. Probabilities of accesses to the activities during the workweek: (a) for internal accesses; (b) for external accesses

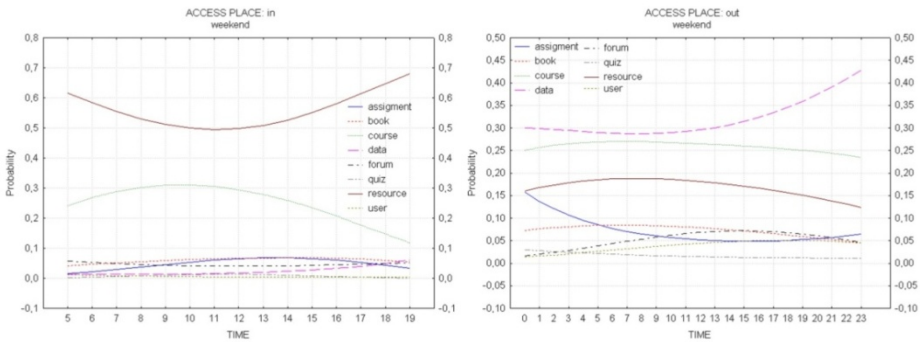


Fig. 4. Probabilities of accesses to the activities during the weekend: (a) for internal accesses; (b) for external accesses

As for the accesses from the university net (internal accesses), the higher probability of selection during the workweek (Fig. 3a) is reached by the activities *quiz, data, course* and *resource*. The probability of selecting *quiz* is small in the morning 0.14 (5.00 a.m.), then it rises, reaching around noon values higher than 0.5, and during the afternoon going down to the value of 0.06 (7.00 p.m.). The activity *resource* shows –an inverse course; the probability of selecting *resource* culminates in the morning and in the evening ($\pi > 0.2$), and during the day goes down ($\pi < 0.1$). A different development can be observed with the activity *data* where the probability of its selection is small during the morning ($\pi < 0.1$), but it rises during the afternoon, reaching the value of 0.46 (7.00 p.m.). The development exactly opposite to the *data* activity can be observed with the *course* activity where the probability of its selection culminates in the morning 0.33 (5.00 p.m.) and during the day goes down to the value of 0.14 (7.00 p.m.).

As during the workweek, also during the weekend (Fig. 4a) the activities *resource* and *course* reach the highest value of selection probability. With the *course* activity, the probability of selection culminates, like during the workweek, in morning hours ($\pi > 0.3$) and during the day goes down to the value of 0.12 (7.00 p.m.).

Approximately the same course as in the workweek can be observed also for the *resource* activity, but with higher values of the probability of selection. The probability of selecting the activity *resource* culminates in the morning ($\pi > 0.61$) and in the evening ($\pi > 0.67$), going down during the day ($\pi < 0.5$).

The accesses from the net outside the university (external accesses) (Fig. 3b; Fig. 4b) show the highest probability of selection for the activities *data, course, resource* and *assignment*. The *data* activity shows little probability of selection during the day, but during the evening it rises to the value of 0.36 (11.00 p.m.) and during the weekend even to the value of 0.43 (11.00 p.m.).

An opposite development can be observed in the activity *assignment* where the probability of selection culminates at night 0.15 (0.00) and goes down during the day. Probabilities of selecting the activities *course* ($\pi < 0.27$) and *resource* ($\pi < 0.17$) are more stable, reaching approximately the same values during the whole week.

5 Evaluation of the Model

The fitness of our models cannot be verified by the deviance, because we get many expected counts $\hat{\mu}_{ii}$ equal to zero. We have provided this in three ways.

First, we compare the probability distribution of the number of empirical accesses to the probability distribution of number of expected accesses to the activities using the Wilcoxon matched pairs test. The test values were not significant for internal and external accesses for all activities as during the workweek, also during the weekend. We can conclude that there is no significant difference between the empirical and the expected counts.

Second, we draw the differences of empirical and theoretical frequencies of accesses to the activities during the workweek for internal accesses and for external accesses (Fig. 5). All means of those differences are equal to zero. Some differences are seen to be large, but if we use "rule of two-times standard deviation" only a few values are outliers (one or two for some activities). Similar results were obtained for the weekend.

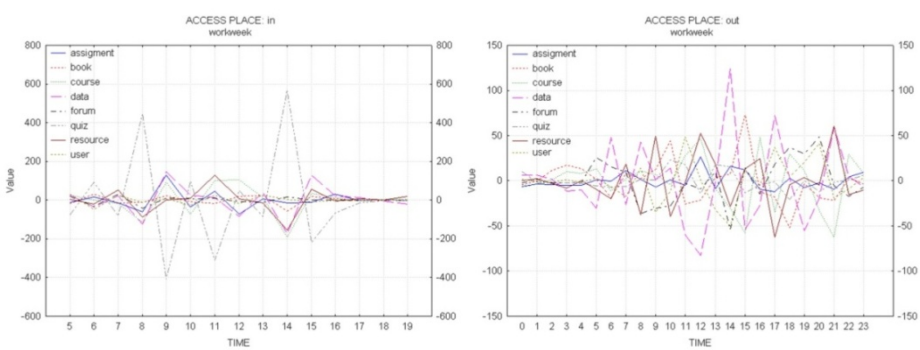


Fig. 5. Differences of empirical and theoretical frequencies of accesses to the activities during the workweek: (a) for internal accesses; (b) for external accesses

Third, we draw the empirical and fitted logits (without the reference category *user*). Figure 6 obtains empirical and theoretical logits during the workweek for selected activities with the highest probabilities for internal accesses and for external accesses. We cannot evaluate the empirical logits in every time *i*, since some counts equal zero. Similar results were obtained for the weekend.

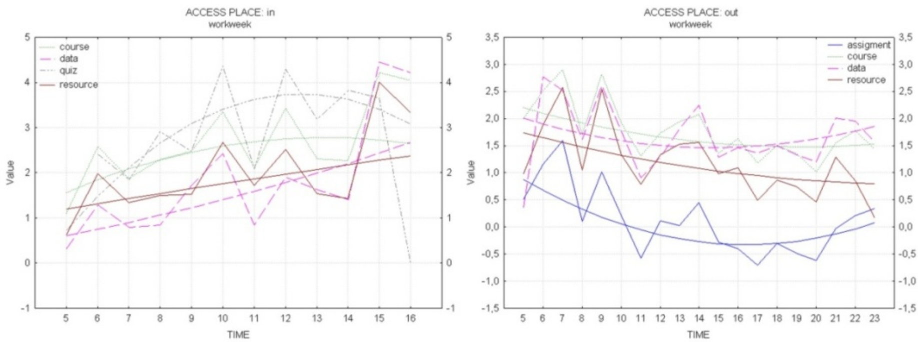


Fig. 6. Empirical and theoretical logits during the workweek: (a) for internal accesses; (b) for external accesses

6 Discussions and Conclusions

The graph (*workweek, in*; Fig. 3a) depicts probabilities of accesses to individual course activities during the day from computers within the university. It confirms several assumptions. At first glance it is evident that the probability of using the *quiz* activity is highest during the workweek. This fact is a result of requirements put on a test – to ensure objectivity, testing is done in a computer room at the university and under the supervision of an instructor. Further expected development shows the probability of using the *book* activity, providing on-line study material, as well as discussion forums. Since discussion forums were used for asynchronous communication in the course and for the dealing with student problems, an almost identical probability of their use during the day is not surprise at all.

The graph also shows several interesting facts. First, it is surprising that some part of students accesses e-learning course in early morning hours (*course* curve), before the first lesson. This probability goes gradually down during the day, from which it may be concluded that after lessons students do not use the university computer rooms to do their assignments, but rather leave the university premises and access e-learning course from their own computers.

Similarly interesting is the highest probability of accessing study resources in the form of off-line materials, downloadable files, etc. (*resource*) outside direct teaching time, be it in the morning or in the afternoon hours. The last activity with an interesting development during the day is the *data*. This includes rather passive activities of students, such as browsing through achieved results and the content of the course – checking grades, test results, achieved points and submitted assignments, checking student resources based on their type, etc. From the graph follows that students are concerned with these activities in later hours of the workday. Other categories of activities are used by students during the work day with the probability lesser than 0.1, therefore their analysis, in our opinion, does not have a practical sense either for the teacher or for the author of the e-learning course.

Putting into relation the graphs depicting probabilities of using activities from within the university net and from without it, during a workday, it is possible to acquire a general view of student behavior in the e-learning course.

It can be noted that the probability of passive browsing through a course content, that is, the activities included in the *data* category, is highest during evening hours (0.4) as well as late evening hours (0.35) (*workweek, out*; Fig. 3b). This finding corresponds with the probability of a "brief look" into the course itself (*course*).

With regard to this probability we were surprised by the fact that the probability of accessing the e-learning course during the workday from outside the university net is almost the same, being the highest in early morning hours. A probable reason for this is the students´ way of study, submission of assignment or preparation for continuous tests at the last moment, etc.

Our assertion is partially documented also by the probability of using the *assignment* activity. As we can see in the graphs, the highest probability of a student´s submission of assignment is around 7 p.m., if he/she is still on the university premises, or then late at night and in early morning hours.

Taking a more detailed look at students´ work with course study materials (*resource*), one can see that it is the use of study resources which is the most frequent activity performed by students accessing the e-learning course from outside the university net in late evening and early morning hours. This is confirmed also by our previous explanation of the unexpectedly high probability of accessing e-learning course in early morning hours.

Relatively surprising is also the following assertion: during the workday there is a higher probability of students´ entering the e-learning course from the outside than from the university net. This fact suggests that in the time between the in-class teaching students leave premises of the university, or they are not present at lessons, despite the fact that they are mostly obligatory. This is an indirect proof that the University has reserves in the creation of conditions and providing complementary services for study within the obligatory activities of students, therefore students leave its premises.

We will provide a short summary of our findings related to student behavior in the monitored e-learning course during weekends. At the beginning it has to be stressed that teaching rules had a direct influence on final probabilities and on their statistical significance. During weekends there was no teaching on the university premises, which were not freely accessible, and this was reflected in the accesses to the course as well. From the graph (*weekend, in*; Fig. 4a) it may be concluded that during the weekend students signed into the course only to take a look or to download study materials.

More relevant findings may be read from the graph "*weekend, out*" (Fig. 4b). As with workdays, also during weekends many students access the e-learning course to find out whether there are some complementary materials, tasks or information related to the study (*course*, *data*). Their activity is higher in evening hours. From the point of view of the teacher, however, we can claim that in this case it is not an active study.

The probability that during the weekend students look for study materials (*resource*) is, during the whole day, higher than 0.1, reaching the maximum in the morning hours. A similarity with the curve characterizing the probability of using on-line sources (*book*) shows that looking for study materials students check this type of study materials as well. Higher values of probability for the category *resource* (and partly also the *book*) are connected with the development of the probability of the *assignment*. As it was said in the discussion regarding the results of study during workdays, it appears that students are willing to work into early morning hours and during weekends if deadline for the submission of assignment is approaching.

The multinomial logit model (MLM) used to assess the probability with which students access individual activities of the e-learning course in individual parts of the day finds its application in various stages of the course´s life cycle, as well as for various roles of LMS users.

It is evident that structure of the e-learning course itself predetermines probability of using individual activities in a particular time of the day, or week. Therefore the greatest contribution does not lie in the process of designing a new course, but rather in the process of restructuring the existing e-learning course. Based on the determined probabilities, the author of the course content may change original representation of individual categories of activities in the course according to the student behavior. If some activity seems to be used only with little probability by students, we can conclude that this way of study was not attractive for them. In such case we should search for alternative activities with the same or similar teaching aim.

The students work with study materials and information sources of the course outside the time of their daily study as it was mentioned in the discussion. Therefore we can say that the results of using MLM may influence not only the structure and content of the e-learning course itself, but also the structure of teaching lessons, e.g. in the combined (blended learning) form of study. The teacher may focus in these lessons on the development of soft skills, use non-traditional forms of work, etc., ensuring the subject content by selecting suitable resources and activities of the course.

Further use of results acquired through the application of MLM can be found in the personalization of the course content – not the content for every user, but rather personalization based on stereotyped users. If, for example, the e-learning course happens to be used simultaneously for internal and external forms of study, it is possible to provide students, based on various probabilities with which they access the course activities at various times and from various places, with their combination as well as with obligatory deadlines for the submission of tasks, assignments, or projects.

Drawing on the determined probabilities, the teacher may, provided that it is allowed by the environment of an individual LMS, adjust teaching management to the needs, or rather habits, of the student target group. It is mainly the selection of suitable deadlines, either for the submission of assignments and tasks, participation in discussion forums, or the realization of tests and self-tests.

This recommendation is also connected with the work of the LMS administrator. From the aspect of the LMS administrator the results acquired through MLM may be used in planning the LMS maintenance. In the discussion we offered a surprising finding that students access the e-learning course with approximately the same probability during the whole day, including the early morning hours. This fact should be taken into consideration during the planning of the LMS maintenance, backing up courses and ensuring an overall accessibility of the system. The system should not be inaccessible when the deadline for the submission of assignments is approaching. Naturally, in searching for a suitable schedule for administrator activities a certain compromise should be attempted and teachers should synchronize deadlines for the submission of assignments in their e-learning courses in order to determine a period with minimum accesses.

References

1. Anděl, J.: Základy matematické štatistiky. MATFYZPRESS, Praha (2007)
2. Rodríguez, G.: Generalized Linear Models (2007)
3. Hosmer, D.W., Lemeshow, S.: Applied Logistic Regression. John Wiley and Sons, Inc., Chichester (2000)
4. Baltagi, B.D.: Econometrics. Springer, Heidelberg (2008)
5. Macfadyen, L.P., Dawson, S.: Mining LMS data to develop an ”early warning system” for educators: A proof of concept. Comput. Educ. 54, 588–599
6. Stratton, L.S., O’Toole, D.M., Wetzel, J.N.: A multinomial logit model of college stopout and dropout behavior. Economics of Education Review 27, 319–331 (2008)
7. Domenech, J.M., Lorenzo, J.: A tool for web usage mining. In: Proceedings of the 8th International Conference on Intelligent Data Engineering and Automated Learning. Springer, Birmingham (2007)
8. Drlik, M., Munk, M., Skalka, J.: Usage Analysis of System for Theses Acquisition and Plagiarism Detection. Procedia Computer Science (2010)
9. Chitraa, V., Davamani, A.S.: A Survey on Preprocessing Methods for Web Usage Data. International Journal of Computer Science and Information Security 7 (2010)
10. Zaine, O.R., Xin, M., Han, J.: Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs. In: Proceedings of the Advances in Digital Libraries Conference. IEEE Computer Society, Los Alamitos (1998)
11. Mor, E., Minguillon, J.: E-learning personalization based on itineraries and long-term navigational behavior. In: Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & posters. ACM, New York (2004)
12. Wei, W., Jui-Feng, W., Jun-Ming, S., Shian-Shyong, T.: Learning portfolio analysis and mining in SCORM compliant environment. In: 34th Annual Frontiers in Education, FIE 2004, vol. 11 (2004), T2C-17-24
13. Talavera, L., Gaudioso, E.: Mining student data to characterize similar behavior groups in unstructured collaboration spaces. In: Workshop on Artificial Intelligence in CSCL, Valencia, Spain, pp. 17–23 (2004)
14. Ramli, A.A.: Web usage mining using apriori algorithm: UUM learning care portal case. In: International Conference on Knowledge Management, Malaysia, pp. 1–19 (2005)

15. Raju, G.T., Satyanarayana, P.S.: Knowledge Discovery from Web Usage Data: a Complete Preprocessing Methodology. IJCSNS International Journal of Computer Science and Network Security 8 (2008)
16. Romero, C., Ventura, S.: Educational data mining: A survey from 1995 to 2005. Expert Systems with Applications 33, 135–146 (2007)
17. Gaudioso, E., Talavera, L.: Data mining to support tutoring in virtual learning communities: Experiences and challenges. In: Romero, C., Ventura, S. (eds.) Data Mining in e-learning, pp. 207–226. Wit Press, Southampton (2006)
18. Ba-Omar, H., Petrounias, I., Anwar, F.: A Framework for Using Web Usage Mining to Personalise E-learning. In: Seventh IEEE International Conference on Advanced Learning Technologies, ICALT 2007, pp. 937–938 (2007)
19. Bayir, M.A., Toroslu, I.H., Cosar, A.: A New Approach for Reactive Web Usage Data Processing. In: Proceedings of the 22nd International Conference on Data Engineering Workshops, p. 44 (2006)
20. Zhang, H., Liang, W.: An intelligent algorithm of data pre-processing in Web usage mining. In: Proceedings of the World Congress on Intelligent Control and Automation (WCICA), pp. 3119–3123 (2004)
21. Cooley, R., Mobasher, B., Srivastava, J.: Data Preparation for Mining World Wide Web Browsing Patterns. Knowledge and Information Systems 1, 5–32 (1999)
22. Yan, L., Boqin, F., Qinjiao, M.: Research on Path Completion Technique in Web Usage Mining. In: International Symposium on Computer Science and Computational Technology, ISCSCT 2008, vol. 1, pp. 554–559 (2008)
23. Yan, L., Boqin, F.: The Construction of Transactions for Web Usage Mining. In: International Conference on Computational Intelligence and Natural Computing, CINC 2009, pp. 121–124 (2009)
24. Spiliopoulou, M., Mobasher, B., Berendt, B., Nakagawa, M.: A Framework for the Evaluation of Session Reconstruction Heuristics in Web-Usage Analysis. INFORMS J. on Computing 15, 171–190 (2003)
25. Munk, M., Kapusta, J., Svec, P.: Data preprocessing evaluation for web log mining: reconstruction of activities of a web visitor. Procedia Computer Science 1, 2273–2280 (2010)
26. Anděl, J.: Matematická štatistika. SNTL, Praha (1985)

Integration of ePortfolios in Learning Management Systems

Ricardo Queirós[1], Lino Oliveira[2], José Paulo Leal[3], and Fernando Moreira[4]

[1] CRACS/INESC-Porto & KMILT/ESEIG/IPP, Porto, Portugal
ricardo.queiros@eu.ipp.pt
[2] KMILT/ESEIG/IPP, Porto, Portugal
linooliveira@eu.ipp.pt
[3] CRACS/INESC-Porto & DCC/FCUP, University of Porto, Portugal
zp@dcc.fc.up.pt
[4] Portucalense University, Porto, Portugal
fmoreira@upt.pt

Abstract. The LMS plays a decisive role in most eLearning environments. Although they integrate many useful tools for managing eLearning activities, they must also be effectively integrated with other specialized systems typically found in an educational environment such as Repositories of Learning Objects or ePortfolio Systems. Both types of systems evolved separately but in recent years the trend is to combine them, allowing the LMS to benefit from using the ePortfolio assessment features. This paper details the most common strategies for integrating an ePortfolio system into an LMS: the data, the API and the tool integration strategies. It presents a comparative study of strategies based on the technical skills, degree of coupling, security features, batch integration, development effort, status and standardization. This study is validated through the integration of two of the most representative systems on each category - respectively Mahara and Moodle.

Keywords: eLearning, ePortfolios, LMS, Interoperability.

1 Introduction

In recent years, eLearning has assumed an important role in schools and companies. The main objective of this educational model is to enhance the teaching/learning process by using the Internet for delivering educational activities. Learning Management Systems (LMS) are specialized systems developed for managing these educational activities, which include the distribution of educational content, the synchronous and asynchronous communication with students and the assessment of students' skills based on assignments and tests. Nevertheless, in order to provide a positive learning experience the LMS must be effectively integrated with other specialized systems typically found in an educational environment such as Repositories of Learning Objects or ePortfolio Systems.

However the growing importance and benefits of ePortfolio systems as a mean for gathering students' achievements and for evaluates student's progress also poses

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 500–510, 2011.

several interoperability issues. For instance, it is important that the evidence of students' work does not disappear, or becomes unusable, when they move to another institution. In this scope, interoperability specifications supported by both ePortfolio systems and LMS are part of the solution.

The goal of this paper is to gather information on how to integrate an ePortfolio system with an LMS. To accomplish it, we identify three integration strategies, namely, the data, the API and the tool integration. Based on these strategies we present a comparative study on ePortfolio interoperability. This study is validated through the integration of two of the most representative of each category - Mahara and Moodle. In this scope three scenarios of integration were explored and for each one we chose a strategy appropriated to its requirements. The outcome of this work should be of interest to anyone defining a strategy for a similar integration.

The remainder of this paper is organized as follows: Section 2 traces the evolution of eLearning with emphasis on ePortfolio systems. In the following section, we detail the different strategies for the integration of ePortfolio Systems in LMS presenting the main advantages and disadvantages of each strategy. Then, we present a case study reflecting the integration of two of the most representative LMS and ePortfolio systems. Finally, we conclude with a summary of the main contributions of this work and a perspective of future research.

2 ePortfolio Systems

An LMS is a software application for the administration, documentation, tracking, reporting of training programs, classroom and online events, and training content [1]. There are open source systems, such as Moodle, Sakai, .LRN or Dokeos, and commercial systems such as WebCT/Blackboard or Desire2Learn. The content delivered by an LMS can be created, obtained, gathered or evaluated in several types of systems such as Learning Objects Repositories, ePortfolio systems, Authoring Tools, Specialized Evaluators or Quizzes.

An ePortfolio is the product created by the student, which contains a collection of digital objects (artifacts), combining various media (audio/video/text/images) [2], articulating experiences, developments, achievements and learning. Its primary aim is to collect evidence for summative assessment, to demonstrate achievement, to record progress and to set targets [3]. The main motivation to integrate an ePortfolio system into an LMS is to use it as an assessment tool.

According to JISC [3], the construction of ePortfolios in the learning process contributes to: (a) improving self-understanding and understanding of the curriculum; (b) engaging and motivating students, both individually and as part of a community of practice; (c) personalizing learning; (d) supporting learning models appropriate to a digital age and (e) promoting reflective practice. These contributions are shared by students, teachers, parents and administrators. For students it shows their accomplishments and encourages them to take responsibility for their work. For teachers it provides a framework for organizing the students' work and facilitates the students' information for assessment and decision making. For parents it offers an insight into what their children do in school. For administrators it provides evidences that teacher/school are being met.

In short, the ePortfolio enables the students to construct a structured collection of their knowledge, skills and competencies [4], allows learners to trace the development of their thinking and learning over time and to show those competencies both to teachers and employers, providing digital resources relevant to their own study (personalised information) and links to other learners (for collaboration and feedback) [5].

Helen C. Barrett [6] organizes the ePortfolio tools in two categories: individual and institutional. Both are presented in Table 1.

Table 1. ePortfolio tools by categories

Individual		Institutional	
Authoring tools	Web Services	Software – Server	Hosted Services
Mozilla Composer	Google Docs	Elgg	Digication
Dreamweaver	Zoho Writer	Mahara	iWebFolio
Microsoft Office	Wikispaces	OSPI	Epsílen
Adobe Acrobat		Moofolio	Goole Aps for Education
Movie Maker		MyStuff	

In the individual category, we can use authoring tools for author portfolios offline (requires web server space to publish online) or web services to create online and publish a presentation portfolio allowing interactivity (Web 2.0). In the institutional category, we can use a software-server approach when an institution uses its own server to provide space for hosting portfolios or hosted services.

In a survey [7] conducted on eLearning systems usage by Portuguese Higher Education Institutions (HEI), no one indicated to be using an ePortfolio system. This fact allows us to conclude that the dissemination of these tools in the educational institutions, at least in Portugal, is still low. This is in part justified due to the lack of standardization of ePortfolio, which renders them difficult to integrate with other systems. In recent years, the development of common standards to represent ePortfolio content, such as IMS LIP [8], IMS ePortfolio [9], Leap2A [10], is being promoted by organizations concerned with ePortfolio interoperability.

The IMS LIP describes the characteristics of a learner, goals and accomplishments. The description is a collection of information about a learner (individual or group learners) or a producer of learning content (creators, providers or vendors). An update (version 1.0.1) was released in early 2005.

The IMS ePortfolio specification was completed in 2004 and represents a profile of existing IMS specifications, namely Content Packaging and Learner Information Package.

The Leap2A is an outcome of the JISC CETIS (JISC Innovation Support Centre - Centre for Educational Technology and Interoperability Standards) project called InterOperability Project. LEAP2A is a simple ATOM based standard for exchanging learning ePortfolio data.

The following table summarizes these standards and for each one we present the support status of the two of the most representative LMS and ePortfolio systems.

Table 2. LMS/ePortfolio support for ePortofilo content standards

	Moodle	Blackboard	Mahara	PebblePad
IMS LIP	Yes	Yes	No	No
IMS e-Portfolio	Yes	Yes	No	Yes
Leap2A	Yes	Yes	Yes	yes

The previous table shows that the ePortfolio standard most used is Leap2A. Beyond Mahara and PebblePAD, other ePortfolio systems support Leap2A such as ePET, MyProgressFile and Passportfolio. On the other hand, the IMS standards due to its complexity and robustness are partially supported.

The Leap2A is a lightweight ePortfolio standard that uses an XML manifest file (leap2a.xml) wrapped with other resources inside a zip file. The manifest is based on Atom syndication format. Atom was designed for exchanging the blog feeds but fits also for exchanging the portfolio information. In the Leap2A, the Atom is extended because Atom's vocabulary is not enough for representing all information stored in ePortfolio systems. The following example shows a XML excerpt of the Leap2A manifest representing a meeting:

```
<entry>
  <title>Agenda</title>
  <id>portfolio:meeting/123</id>
  <updated>2007-11-19T01:00:00Z</updated>
  <content type="text">
    Meeting with John
  </content>
  <link rel="leap:is_agenda_of" href="portfolio:meeting/45"/>
  <rdf:type rdf:resource="leaptype:entry" />
</entry>
```

Information in Leap2A is grouped into items, each represented as an Atom entry with additional LEAP specific metadata. Each item has a Leap2A type or class, and the type affects which literal attributes, relationships or categories that may be associated with the item [10].

These ePortfolio standards are widely used in several interoperability specifications [11, 12, 13] and projects [14, 15, 16]. One of such projects is PEACE (Project for ESEIG Academic Content Environment). This project aims to integrate an ePortfolio system (Mahara) with the institution's LMS (Moodle) as part of a learning environment composed by several services targeting the new Web 2.0 paradigm [4, 16].

3 Integration Strategies

In this section we present the most common strategies for integrating an ePortfolio system into an LMS, namely the Data, the API and the Tool integration strategies.

3.1 Data Integration

Data integration is the simplest and most popular form of integration in content management. This type of integration uses the import/export features of both systems and relies on the support of common formats as shown in Fig. 1. For instance, an ePortfolio system may import data (blog and forum contributions by students, course materials and assignments uploaded by teachers) from LMS to avoid the burden of entering this data manually.

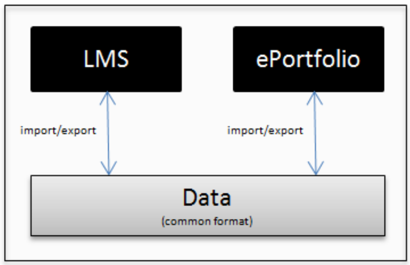


Fig. 1. Data integration

These systems support two types of common formats: generic (e.g. HTML files) and ePortfolio specific (e.g. Leap2A files). The former are useful since they are widely available, but they lack domain specific semantic data provided by the latter. For instance, if we add a post in a Moodle forum it should be included in the Mahara ePortfolio as a blog post and not as a non-editable HTML artifact. This requires the use of a common ePortfolio standard so that Mahara (or any other ePortfolio system) can understand the meaning of the content and decides its final format.

3.2 API Integration

An Application Programming Interface (API) allows client applications to use directly the functions of an eLearning system. These APIs foster client application development through data encapsulation and behavioral reuse. This clear separation of interfaces specification from their implementation and data formats allows tool vendors to develop new versions without affecting current clients [17].

The major LMS vendors include APIs to allow developers to extend their predefined features through the creation of plugins. Blackboard uses the Building Blocks technology to cover the integration issues with other systems allowing third parties to develop modules using the Building Blocks API.

The new Moodle version (v. 2.0 released in November 2010) includes several APIs (Fig. 2) to enable the development of plugins by third parties to access repositories and portfolios such as the **Repository API** for browsing and retrieving files from external repositories; and the **Portfolio API** for exporting Moodle content to external repositories.

These two APIs are based on the File API - a set of core interfaces to allow Moodle to manage access control, store and retrieve files. The new File API aims to enhance file handling by avoiding redundant storage. This feature is achieved since every file in Moodle 2.0 is saved into a file pool (a directory in `moodledata`) with a filename that is calculated as a SHA1 hash of the file content. If a file is copied (e.g. course cloning) no file duplication happens, just a new record in a special table of files is created.

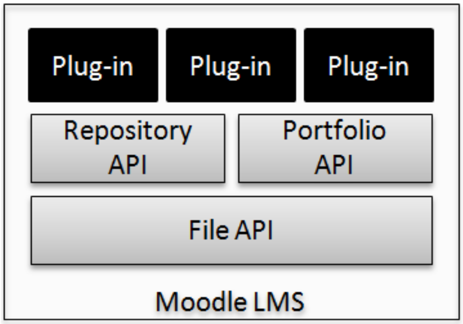


Fig. 2. Moodle File API

In order to ensure a bidirectional communication between a LMS and an ePortfolio system it is required to use both APIs to create plugins. For instance, in the Moodle LMS, the Mahara support is guaranteed only in one way by the implementation of the Portfolio API.

The Portfolio API is a core set of interfaces to publish files from Moodle to external repository systems, mainly ePortfolio systems. In this approach, the ePortfolio system appears seamlessly as a folder when students want to save content such as a file, snapshots of forums or blogs and assignments.

At time of writing this paper, Moodle 2.0.1 (January 2011) includes in its release package several plugins for ePortfolios such as Mahara, Flickr, Google Docs, Boxnet and supporting different formats such as Leap2A and HTML. Regarding the Repository API the same release package includes support for the repositories Alfresco, Boxnet, Dropbox, Flickr, Google Docs, Merlot and Picasa.

3.3 Tool Integration

The IMS Learning Tools Interoperability (IMS LTI) provides a uniform standards-based extension point in LMS allowing remote tools and content to be integrated into LMSs. The main goal of the LTI is to standardize the process for building links between learning tools and the LMS. There are several benefits from using this approach: educational institutions, LMS vendors and tool providers by adhering to a clearly defined interface between the LMS and the tool, will decrease costs, increase options for students and instructors when selecting learning applications and also potentiate the use of software as a service (SaaS). The LTI has 3 key concepts as shown in Fig. 3 [18]: the Tool Provider, the Tool Consumer and the Tool Profile.

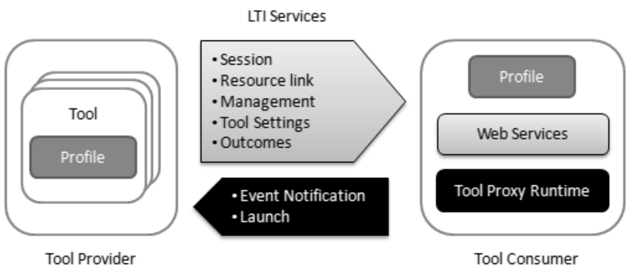


Fig. 3. IMS Full LTI

The *tool provider* is a learning application that runs in a container separate from the LMS. It publishes one or more tools through tool profiles. A *tool profile* is an XML document describing how a tool integrates with a tool consumer. It contains tool metadata, vendor information, resource and event handlers and menu links. The *tool consumer* publishes a Tool Consumer Profile (XML descriptor of the Tool Consumer's supported LTI functionality that is read by the Tool Provider during deployment), provides a Tool Proxy Runtime and exposes the LTI services.

A subset of the full LTI v1.0 specification called IMS Basic LTI exposes a single (but limited) connection between the LMS and the tool provider. For instance, there is no provision for accessing run-time services in the LMS and only one security policy (OAuth protocol [19]) is supported.

For instance, to export content from Moodle to Mahara using the Basic LTI the teacher (or LMS administrator) must first configure the tool (Mahara) as a Basic LTI tool in the course structure. When a student selects this tool, Moodle launches a Mahara session for the student. The web interface for this session can either be embed in Moodle’s web interface as an *iframe* or launched in a new browser window.

3.4 Comparison of the Integration Strategies

In this subsection we present a comparative study on the ePortfolio integration strategies in LMS. This study is summarized in Table 3 and can be used as a guide in the selection of an integration strategy.

Table 3. Comparison of ePortfolio integration strategies

	Data Integration	API Integration	Tool Integration	
			bLTI	fLTI
Technical skills	No	Yes	Yes	Yes
Degree of coupling	No bounding	Tightly	Loosely	Tightly
Security	To implement	To implement	OAuth	
Batch integration	No	Yes	No	
Development effort	-	Some	Little	Great
Communication type	Bidirectional	Bidirectional	Uni	Bi
Status (# implementations)	-	Many	Many	Few
Standards	Web/image/video formats	Leap2A, HTML	Leap2A, HTML	

Data integration is the best option when the development effort must be kept to a minimum or no one with technical skills (specially programming skills) is available, since the other two strategies require them. This strategy has also the advantage of not coupling the two systems and enabling a bi-directional communication.

API integration is best suited when batch integration is required since the other two strategies involve the use of the GUI of both systems. For instance, if the work of the students of a given set of courses must be copied on a regular basis from the LMS to their portfolios then the API strategies are recommended. The major drawbacks of this approach are the amount of development required and the tight coupling between the LMS and the ePortfolio system, since special plugins must be implemented and APIs are vendor specific. Finally, this strategy enables bidirectional communication, although the current version of Moodle (2.0.1) does not implement yet the API repository, thus rendering in practice the communication between LMS and Mahara unidirectional.

Tool integration is arguably the best choice in general since it provides a good balance between implementation effort and coupling and security. This is especially true if only unidirectional communication is required and Basic LTI is used. This tool integration flavor is simple to implement and is already supported by most LMS vendors. If bidirectional communication is required then full LTI is needed but in this case the implementation is harder and few LMS vendors support this flavor of the specification. In both cases, tool integration has the added value of providing some basic security features based on the OAuth protocol aiming to secure the message interactions between the Tool Consumer and the Tool Provider.

This comparative study was based both on the available documentation and on the authors experience in using the different strategies to integrate Moodle with other systems, in particular the development of a Moodle plugin using the Repository API [20] and the basic LTI runtime to link Moodle with other eLearning systems [21].

4 Case Study

This section presents an example of integration of an ePortfolio system into Moodle LMS as part of PEACE (Project for ESEIG Academic Content Environment), an ongoing project of the School of Industrial Studies and Management (ESEIG) of the Polytechnic Institute of Porto (IPP) [16].

This case study is related to the *social part* of the PEACE project that implements a platform for creating students' controlled personal learning networks (PLN) integrating their personal learning environments (PLE), ePortfolios, Web 2.0 (applications, services and people interactions) and LMS, as shown in Fig. 4 [4]. This environment is developed using an open source application to create ePortfolios and social networks – Mahara – integrated with the Moodle LMS.

Mahara and Moodle share the identity of the student via Lightweight Directory Access Protocol (LDAP) so that a student has a single set of authentication data (login and password) on both applications (Single Sign-On). Thus, a student authenticated in Moodle can automatically access his/her Mahara profile without needing to login again. If the student has not yet created a profile in Mahara, it is created automatically on first access based on data from his/her Moodle profile.

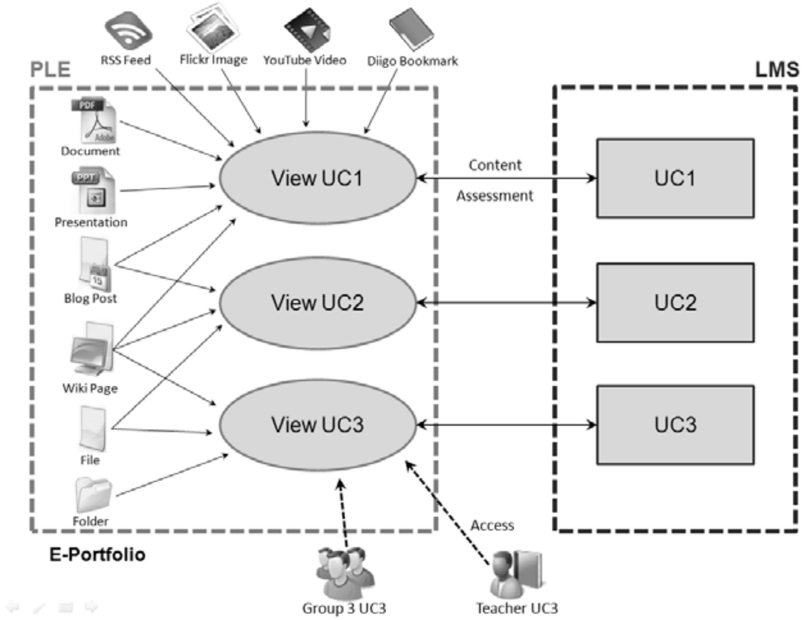


Fig. 4. Integration of ePortfolio, PLE and LMS [4]

We identified three scenarios where the integration between both systems is relevant. For each scenario we chose one of the strategies presented in the previous section according to the requirements of that particular scenario. These scenarios are:

1. the manual and sporadic copy of a single resource from Moodle to Mahara (or vice-versa);
2. the automatic and periodic export of new students work from Moodle to their Mahara accounts;
3. the visualization of a Mahara portfolio from a Moodle user profile.

In the first scenario a student needs to export a single resource (e.g. a work in the PDF format) from Moodle to a specific view in his Mahara portfolio. Given the sporadic nature of this use case, we selected the data integration strategy. In this strategy files sharing is based on the import/export features of both systems. In this case, the student exports the resource in its native format from Moodle and imports the same resource in Mahara. Both systems support a wide range of file formats such as HTML, PDF, image and video formats.

In the second scenario, the goal is to implement a mechanism to periodically (e.g. annually) export the work of all students to their respective portfolios to have their academic record in Mahara. The purpose of this export is twofold: to have the academic work produced in Moodle organized by student rather than by course, and to incentive students to maintain their own portfolios by populating them with work they have done in Moodle. In this scenario, we must use batch integration since any manual/graphical interaction would be time-consuming and error-prone. This is a clear case for the API integration strategy since this is the only one that supports batch communication. This is achieved by overriding the methods `prepare_package` and `send_package` of the `portfolio_plugin_base` class included in the Portfolio API. The former prepares the package for sending, writing out a metadata manifest file and zipping all the files in a temporary folder. The latter sends the package to a remote system based on a XMLRPC request. A script calling these methods in sequence is invoked periodically by `cron` - a system command that runs predefined tasks on a computer at regular intervals.

In the third scenario, the goal is to present the portfolio of a student embedded in the Moodle profile. Tool integration is the ideal strategy for this scenario, although in the near future this approach will provide only partial integration since none of the systems supports full IMS LTI. The current version of Moodle supports only IMS Basic LTI that is currently being developed also for Mahara. This approach gives to the student a perception that only one system is running since the Mahara view is embedded in Moodle's graphical interface avoiding the need to open both systems.

5 Conclusion

This paper presents and compares three strategies to integrate an ePortfolio system into an LMS, namely the data, the API and the tool integration strategies. The comparison is based on the technical skills, degree of coupling, security, batch integration, development effort, communication type, status and standardization, required by each strategy. This study is validated through a case study conducted in the scope of the PEACE project that aims to implements a platform for student controlled personal learning networks. In this study, we particularized on the integration of two of the most representative LMS and ePortfolio systems – respectively Moodle and Mahara. Three scenarios of integration where explored; for each one we chose a strategy appropriated to its requirements.

The main contribution of this work is a survey of the most popular integration strategies currently available for the systems under consideration, and criteria for selecting the most appropriated for a given situation. Although the survey and the test case focused on ePortfolio systems and LMSs, many of issues and solutions discussed here can be adapted to other types of eLearning systems. Thus, we expect this paper to be of interest to anyone concerned with eLearning system interoperability.

As part of the PEACE project, our plans include the support for bidirectional communication between Moodle and Mahara, as soon as the Repository API for Mahara is available. This will enable to browse and to retrieve files from Mahara and integrate them into Moodle resources. In a near future, we also plan to implement batch integration to automate the copy of Moodle students' work to their respective portfolios.

References

1. Ellis, R.K.: Field Guide to Learning Management Systems, ASTD Learning Circuits (2009)
2. Barret, H.: Electronic Portfolios in STEM - What is an Electronic Portfolio, http://www.scribd.com/doc/40206175/E-Portfolio-Definition

3. JISC, Effective Practice with e-Portfolios: Supporting 21st century learning, http://www.jisc.ac.uk/media/documents/publications/effectivepracticeeportfolios.pdf
4. Oliveira, L., Moreira, F.: Personal Learning Environments: Integration of Web 2.0 Applications and Content Management Systems. In: Proceedings of 11th European Conference on Knowledge Management (ECKM 2010), vol. 2, pp. 1171–1177 (2010); Presented at ECKM 2010, Universidade Lusíada de Vila Nova de Famalicão, Famalicão, Portugal
5. Malita, L.: E-portfolios in an educational and occupational context. Procedia - Social and Behavioral Sciences 1(1), 2312–2316 (2009), doi:10.1016/j.sbspro.2009.01.406
6. Barret, H.C.: Categories of ePortfolio Tools, Technical Report, http://electronicportfolios.com/categories.html
7. Queirós, R., Pinto, M., Rodrigues, A.C., Babo, R., Lopes, A., Coelho, P.: Differences in Internet and LMS usage - A Case Study in Higher Education. In: Chapter of the book called "Higher Education Institutions and Learning Management Systems: Adoption and Standardization. IGI Global (to be published, 2011)
8. IMS Learner Information Package (LIP) - Version 1.0.1 Final Specification, http://www.imsglobal.org/profiles/lipv1p0p1/imslip_sumcv1p0p1.html
9. IMS ePortfolio Specification, http://www.imsglobal.org/ep/
10. Leap2A Specification, http://wiki.cetis.ac.uk/LEAP2A_specification
11. Moodle Repository API, http://docs.moodle.org/en/Development:Repository_API
12. Moodle Portfolio API, http://docs.moodle.org/en/Development:Portfolio_API
13. IMS Basic Learning Tools Interoperability Specification – V. 1.0 Final Specification (2010), http://www.imsglobal.org/lti/blti/bltiv1p0/ltiBLTIimgv1p0.html
14. The Portfolio Interoperability Prototyping Project (PIOP), http://wiki.cetis.ac.uk/Portfolio_interoperability_prototyping
15. Alario, C., Wilson, S.: Comparison of the main alternatives to the integration of external tools in different platforms. In: ICERI 2010 Proceedings, pp. 3466–3476 (2010)
16. Queirós, R., Oliveira, L., Pinto, M., Silva, C.: Towards e-Learning 2.0: Case Study of an e-Learning Environment. In: Proceedings of the 9th European Conference on e Learning (ECEL 2010), pp. 812–815. Instituto Superior de Engenharia do Porto, Portugal (2010)
17. Amsdem, J.: Levels of Integration – Five ways you can integrate with the Eclipse Platform, http://www.eclipse.org/articles/Article-Levels-Of-Integration/levels-of-integration.html
18. Gilbert, T., Pearson Education: Leveraging Sakai and IMS LTI to standardize integrations. In: 10th Sakai Conference, Boston, Mass (2009)
19. OAuth security protocol, http://oauth.net/
20. Leal, J.P., Queirós, R.: Integration of Repositories in eLearning Systems. In: ICEIS 2010 - 12th International Conference on Enterprise Information Systems, Madeira, Portugal, pp. 10–12 (2010)
21. Leal, J.P., Queirós, R.: Using the Learning Tools Interoperability Framework for LMS Integration in Service Oriented Architectures. In: TECHEDUCATION 2011 – 2nd International Conference on Technology Enhanced Learning, Quality of Teaching and Reforming Education, Corfu, Greece (May 2011) (to appear)

Monitoring and Control in a Spatially Structured Population Model

Manuel Gámez[1], Inmaculada López[1,⋆], József Garay[2], and Zoltán Varga[3]

[1] Department of Statistics and Applied Mathematics,
University of Almería, La Cañada de San Urbano, 04120 Almería, Spain
{mgamez,milopez}@ual.es

[2] Research Group of Theoretical Biology and Ecological Modelling
of Hungarian Academy of Science and
Department of Plant Taxonomy and Ecology, L. Eotvos University
Pázmány P. st1/c,H-1117 Budapest, Hungary
garayj@ludens.elte.hu

[3] Institute of Mathematics and Informatics, Szent István University,
Páter K. u. 1., H-2103 Godollo, Hungary
Varga.Zoltan@gek.szie.hu

Abstract. In the paper methods of Mathematical Systems Theory are applied to the dynamical analysis of a harvested population with a reserve area. Although the methodology also applies to rather general spatially structured populations, for a concrete interpretation, we consider a fish population living in a free fishing area and in a reserved area, with migration between them. Using a fishing effort model based on logistic growth in both areas, from the catch, by the construction of an auxiliary system called observer, we dynamically estimate the total fish stock. A similar method also applies to the case of a changing environment, when there is a time-dependent abiotic environmental effect described by an additional exosystem. Furthermore, we also consider the problem of steering the population into a desired new equilibrium. To this end an optimal control problem is set up, which is numerically solved using an optimal control toolbox developed for MatLab.

Keywords: spatially structured population, fishing effort model, observer system, equilibrium control.

1 Introduction and Basic Dynamic Model

Conservation ecology is obviously interested both in the maintenance of a biologically valuable ecosystem and the re-establishing a desired equilibrium, by means of human intervention or treatment, see e.g. [11]; while management of renewable natural resources ([2]) is mainly aimed at optimal sustainable harvesting. Monitoring of a population system (estimation of the state from certain

⋆ Corresponding author.

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 511–520, 2011.

measured components) and its control by harvesting are natural questions. Based on the initiative of [14], [15], in recent years, for the state estimate of certain multispecies systems local observers and equilibrium control have been constructed, see e.g. [9], [10], [4], and the survey paper [16]. For a quick review of different observer construction methods we also refer to [5]. In optimal harvesting models, time-dependent fishing effort has been used as control function, see e.g. [2], [12], [3]. In the present paper harvesting is used to control the system into a required equilibrium. Below, from [3], we recall the dynamics of the fish population moving between two zones, consisting of reserved and unreserved areas. In Section 2 monitoring problems of the fishing effort model with reserve area are considered, Section 3 is dedicated to controlling the fish population to a desired equilibrium, and followed by a Discussion section.

Let $x_1(t)$ and $x_2(t)$ be the biomass densities of the same fish population inside a free fishing area and a reserve area, respectively, at a time t. We suppose the population growth is logistic in each area, the fish subpopulation of the free area migrate into reserve area at rate m_{12} and the inverse migration rate is m_{21}. Let E be the *fishing effort* applied in the free area. Then dynamics of the population system can be described by equations

$$\begin{aligned} \dot{x_1} &= r_1 x_1 \left(1 - \frac{x_1}{K_1}\right) - m_{12}x_1 + m_{21}x_2 - qEx_1 \\ \dot{x_2} &= r_2 x_2 \left(1 - \frac{x_2}{K_2}\right) + m_{12}x_1 - m_{21}x_2 \ . \end{aligned} \tag{1}$$

Where r_1 and r_2 are the intrinsic growth rates, K_1 and K_2 the carrying capacities in the free and reserve areas, respectively; and q is the catchability coefficient. From [3] we recall that conditions

$$\frac{r_2(r_1 - m_{12} - qE)^2}{K_2 m_{21}} < \frac{(r_2 - m_{21})r_1}{K_1} \tag{2}$$

$$(r_2 - m_{21})(r_1 - m_{12} - qE) < m_{12}m_{21} \tag{3}$$

$$\frac{r_1 x_1^*}{K_1} > r_1 - m_{12} - qE \ , \tag{4}$$

imply the existence of a unique positive equilibrium $x^* = (x_1^*, x_2^*)$. Conditions (2)-(4) will be supposed throughout the paper.

Example 1. For an illustration, we consider system (1) with the same model parameters as [5]. In Figure 1 we can observe how different trajectories of the model tend to the only equilibrium x^*.

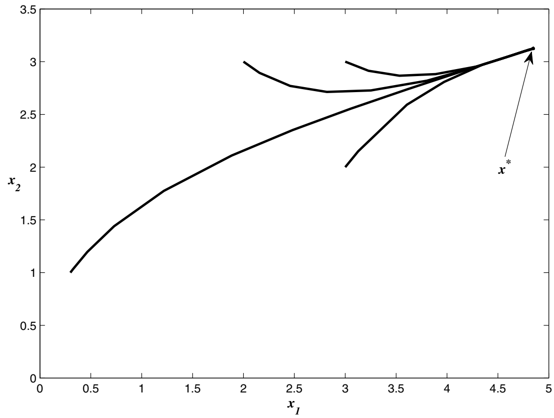


Fig. 1. Some solutions of system (1)

2 Monitoring of the Fishing Effort Model with Reserve Area

2.1 Stock Estimation from the Harvested Fish

First we consider the problem of estimation of the stock in the reserve area, on the basis of the biomass harvested in the free area. (For technical reason its difference from is equilibrium value is considered.) Now, the observation equation

$$y = h(x) = qE(x_1 - x_1^*) \ .$$

Then

$$C := h'(x^*) = (\, qE \quad 0 \,) \ ,$$

and linearizing the model (1), for the Jacobian of the right- hand side we obtain

$$A := \begin{bmatrix} r_1 - 2r_1 \dfrac{x_1^*}{K_1} - m_{12} - qE & m_{21} \\ \\ m_{12} & r_2 - 2r_2 \dfrac{x_2^*}{K_2} - m_{21} \end{bmatrix} .$$

It is easy to check that $rank[C|CA]^T = 2$. Thus, by Theorem 1 of Appendix the system is locally observable near the equilibrium in the sense of Definition 1 of Appendix, and the whole system state (in particular the stock of the species in the reserve area) can be recovered, observing the biomass harvested per unit time. Applying the theory of [13] we can construct a corresponding observer system, as illustrated in

Example 2. As a numerical example, for a comparison of the results we consider the same model parameters as [5]:

$$\begin{aligned}\dot{x_1} &= 0.7x_1\left(1-\frac{x_1}{10}\right) - 0.2x_1 + 0.1x_2 - 0.25\cdot 0.9x_1 \\ \dot{x_2} &= 0.5x_2\left(1-\frac{x_2}{2.2}\right) + 0.2x_1 - 0.1x_2 \ .\end{aligned} \tag{5}$$

Now $x^* = (4.85, 3.12)$ and with

$$K := \begin{pmatrix} 0 \\ 10 \end{pmatrix} ,$$

matrix $A - KC$ is Hurwitz, therefore by Theorem 2 of Appendix we have the following observer system

$$\begin{aligned}\dot{z_1} &= 0.7z_1\left(1-\frac{z_1}{10}\right) - 0.2z_1 + 0.1z_2 - 0.25\cdot 0.9z_1 \\ \dot{z_2} &= 0.5z_2\left(1-\frac{z_2}{2.2}\right) + 0.2z_1 - 0.1z_2 + 10[y - 0.25\cdot 0.9(z_1 - x_1^*)] \ .\end{aligned} \tag{6}$$

If we take an initial condition $x^0 := (30, 120)$ for system (5), and similarly, we consider another nearby initial condition, $z^0 = (35, 100)$ for the observer system (6), then the corresponding solution z of the observer tends to the solution x of the original system, as shown in Figure 2. We note that the convergence is much quicker than that of the observer constructed in [5].

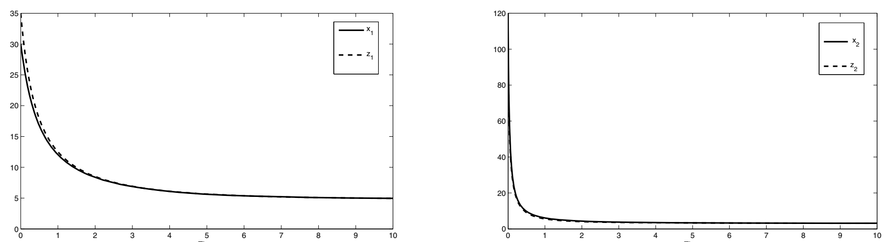


Fig. 2. Solution of observer (6), converging to the solution of system (5)

2.2 Observer for a System with Exosystem Describing Environmental Change

In this section we will design observer for a fishery model, supposing that in the abiotic environment there is a continuous change obeying a known dynamic law, which affects certain parameters of the fishery model. This abiotic process may be e.g. pollution produced by an industrial plant, a periodical (seasonal) change of temperature, or a monotonous increase of the mean temperature due to global warming, etc. In our illustrative example below, this dynamics will be described by simple differential equations, assuming that, with known positive constants α and δ, the external system

$$\begin{aligned} \dot{w}_1 &= \alpha w_2 \\ \dot{w}_2 &= -\delta w_1 \ , \end{aligned} \tag{7}$$

describes a periodic change that affects the coefficients of the diffusion between the two zones in the following form

$$\begin{aligned} \dot{x_1} &= r_1 x_1 \left(1 - \frac{x_1}{K_1}\right) - (m_{12} + c_{12} w_1) x_1 + (m_{21} + c_{21} w_1) x_2 - qE x_1 \\ \dot{x_2} &= r_2 x_2 \left(1 - \frac{x_2}{K_2}\right) + (m_{12} + c_{12} w_1) x_1 - (m_{21} + c_{21} w_1) x_2 \\ \dot{w}_1 &= \alpha w_2 \\ \dot{w}_2 &= -\delta w_1 \ , \end{aligned} \tag{8}$$

where all c_{ij} -s are positive. Since zero is a Lyapunov stable equilibrium of system (7), it is not hard to see that equilibrium $(x^*, 0)$ of system (8) is also Lyapunov stable. With the same observation

$$y = h(x) = qE(x_1 - x_1^*) \ ,$$

as in the previous examples, and applying Theorem 3, we can construct an observer for this system.

Example 3. We consider the system

$$\begin{aligned} \dot{x_1} &= 0.7 x_1 \left(1 - \frac{x_1}{10}\right) - (0.2 + 0.2 w_1) x_1 + (0.1 + 0.1 w_1) x_2 - 0.25 \cdot 0.9 x_1 \\ \dot{x_2} &= 0.5 x_2 \left(1 - \frac{x_2}{2.2}\right) + (0.2 + 0.2 w_1) x_1 - (0.1 + 0.1 w_1) x_2 \\ \dot{w}_1 &= 1.3 w_2 \\ \dot{w}_2 &= -2.1 w_1 \ . \end{aligned} \tag{9}$$

Now using matrix

$$K := \begin{pmatrix} 0 \\ 10 \end{pmatrix} \ ,$$

we obtain the following observer

$$\begin{aligned}
\dot{z_1} &= 0.7z_1\left(1-\frac{z_1}{10}\right) - (0.2+0.2w_1)z_1 + (0.1+0.1w_1)z_2 - 0.25\cdot 0.9z_1\\
\dot{z_2} &= 0.5z_2\left(1-\frac{z_2}{2.2}\right) + (0.2+0.2w_1)z_1 - (0.1+0.1w_1)z_2 + [y-0.225(z_1-x_1^*)]10\\
\dot{w}_1 &= 1.3w_2\\
\dot{w}_2 &= -2.1w_1 \; .
\end{aligned} \tag{10}$$

For the solution of the composite system (9), we set initial value $(x^0, w^0) := (30, 120, 0.1, 0.1)$, and calculate the solution of the observer system (10) with initial value $z^0 := (40, 40)$. Figure 3 shows how the state process is approximated by the solution of the observer even in case of variable environment.

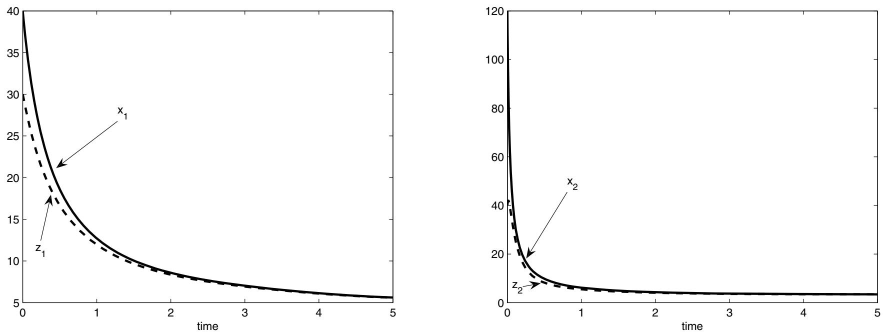


Fig. 3. Solution of observer (10), converging to the solution of system (9)

3 Controlling the Fish Population to a Desired Equilibrium

Let us suppose first that the fish population is controlled by a time-dependent fishing effort of the form $E + u(t)$, where u is defined on a fixed time interval $[0, T]$. Then from model (1) we obtain the control system

$$\begin{aligned}
\dot{x_1} &= r_1x_1\left(1-\frac{x_1}{K_1}\right) - m_{12}x_1 + m_{21}x_2 - q(E+u(t))x_1\\
\dot{x_2} &= r_2x_2\left(1-\frac{x_2}{K_2}\right) + m_{12}x_1 - m_{21}x_2 \; .
\end{aligned} \tag{11}$$

In terms of the notation of the Appendix, with $F : \mathbb{R}^3 \to \mathbb{R}^2$,

$$F(x_1, x_2, u) := \begin{bmatrix} r_1 x_1 \left(1 - \dfrac{x_1}{K_1}\right) - m_{12}x_1 + m_{21}x_2 - q(E + u(t))x_1 \\ \\ r_2 x_2 \left(1 - \dfrac{x_2}{K_2}\right) + m_{12}x_1 - m_{21}x_2 \end{bmatrix},$$

control system (11) takes the form

$$\dot{x} = F(x, u^* + u(t)) \, . \tag{12}$$

Obviously, to $u^* := 0$ and $u(t) := 0$ ($t \in [0, T]$), there corresponds the non-trivial ecological equilibrium x^* of dynamic system (1). For control system (12) it can be proved ([8]) that a rank condition analogous to that of Theorem 1 implies local reachability of (12) from x^* (in time T). The latter means that, using an appropriate control, the system can be steered from to any point of a neighbourhood of x^*. In fact, for a given y^*, we will calculate a control function (time-dependent fishing effort) such that for the corresponding solution x the distance between $x(T)$ and y^* is minimal. This optimal control problem can be numerically solved using an optimal control toolbox developed for MatLab in [1] and [6]. Once the system is in state y^*, with an appropriate constant fishing E_1, this new equilibrium can be maintained, as illustrated in the following

Example 4. As a numerical example, we consider the same parameters of Example 2:

$$\begin{aligned} \dot{x_1} &= 0.7x_1 \left(1 - \frac{x_1}{10}\right) - 0.2x_1 + 0.1x_2 - 0.25 \cdot 0.9x_1 \\ \dot{x_2} &= 0.5x_2 \left(1 - \frac{x_2}{2.2}\right) + 0.2x_1 - 0.1x_2 \, . \end{aligned} \tag{13}$$

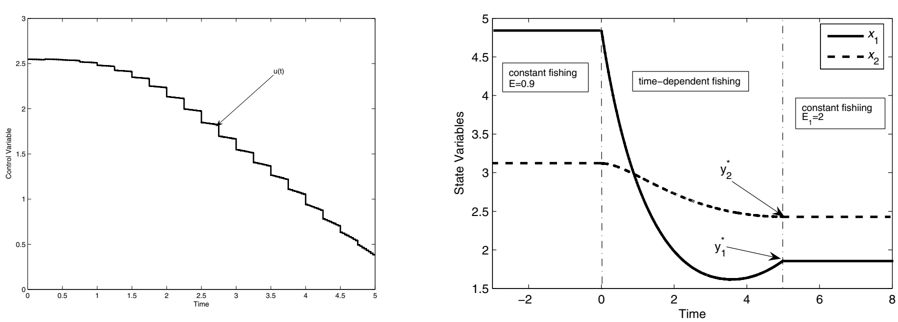


Fig. 4. Control function and solution of system (11) for $T = 5$, with initial value $x(0) = (4.85, 3.12)$

We know that, the system (13) has a nonnegative equilibrium: $x^* = (4.85, 3.12)$, corresponding to a constant fishing effort $E = 0.9$, and we want to steer the system in the new equilibrium $y^* = (1.86, 2.43)$ by the time moment $T = 5$. The obtained results are shown in Figure 4.

4 Discussion

The monitoring methodology based on observer design can be applied to stock estimation in spatially structured fish populations, in particular, when in a reserve area fishing is prohibited.

A further development of the Luenberger observer by [13], dealing with an exosystem describing time-variation of certain parameters made it possible to deal population models in changing environment.

Although our observer is local, in comparison with the global observer of [5] constructed in a different way, in the considered case performs better (converging more quickly).

The problem of steering the spatially structured population into a desired equilibrium can be reduced to an optimal control problem that can be solved numerically applying a toolbox developed for MatLab.

Finally, our methodology can be extended to certain cases of density-dependent migration between the two areas, and also to predator-prey systems with reserve area for the prey, see e.g. [7].

Acknowledgments. The research has been supported by the Scientific and Technological Innovation Fund (of Hungary, ES-17/2008), the Ministry of Education and Sciences (of Spain, HH2008-0023) and the Hungarian Scientific Research Fund OTKA K62000, K81279.

References

1. Banga, J.R., Balsa-Canto, E., Moles, C.G., Alonso, A.A.: Dynamic Optimization of Bioprocesses: Efficient and Robust Numerical Strategies. Journal of Biothecnology 117, 407–419 (2005)
2. Clark, C.W.: Mathematical Bioeconomics: The Optimal Management of Renewable Resources. Wiley, New York (1990)
3. Dubey, B., Chandra, P., Sinha, P.: A model for Fishery Resource with Reserve Area. Nonlinear Analysis. Real World Applications 4, 625–637 (2003)
4. Gámez, M., López, I., Shamandy, A.: Open- and Closed-Loop Equilibrium Control of Trophic Chains. Ecological Modelling 221(16), 1839–1846 (2010)
5. Guiro, A., Iggidr, A., Ngom, D., Touré, H.: On the Stock Estimation for Some Fishery Systems. Review in Fish Biology and Fisheries 19, 313–327 (2009)
6. Hirmajer, T., Balsa-Canto, E., Banga, J.R.: DOTcvpSB, a Software Toolbox for Dynamic Optimization in Systems Biology. BMC Bioinformatics 10, 199 (2009)
7. Khamis, S.A., Tchuenche, J.M., Lukka, M., Heiliö, M.: Dynamics of fisheries with prey reserve and harvesting. International Journal of Computer Mathematics (in press, 2011), doi: 10.1080/00207160.2010.527001

8. Lee, E.B., Markus, L.: Foundations of Optimal Control Theory. Wiley, New York (1971)
9. López, I., Gámez, M., Molnár, S.: Observability and Observers in a Food Web. Applied Mathematics Letters 20(8), 951–957 (2007a)
10. López, I., Gámez, M., Garay, J., Varga, Z.: Monitoring in a Lotka-Volterra Model. BiosSystems 83, 68–74 (2007b)
11. Pimm, S.L.: The Balance of Nature? Ecological Issues in the Conservation of Species and Communities. University of Chicago Press (1991)
12. Shamandy, A., Varga, Z.: An Age-Specific Optimal Harvesting Model. Pure Mathematics and Applications 8(1), 101–110 (1997)
13. Sundarapandian, V.: Local Observer Design for Nonlinear Systems. Mathematical and Computer Modelling 35, 25–36 (2002)
14. Varga, Z.: On Controllability of Fisher's Model of Selection. In: Dafermos, C.M., Ladas, G., Papanicolau, G. (eds.) Differential Equations, pp. 717–723. Marcel Dekker, New York (1989)
15. Varga, Z.: On Observability of Fisher's Model of Selection. Pure Math. and Appl. Ser. B 1, 15–25 (1992)
16. Varga, Z.: Applications of Mathematical Systems Theory in Population Biology. Periodica Mathematica Hungarica 56(1), 157–168 (2008)

Appendix

Given positive integers m, n, let

$$f : \mathbb{R}^n \to \mathbb{R}^n, \quad h : \mathbb{R}^n \to \mathbb{R}^m \ ,$$

be continuously differentiable functions and for some $x^* \in \mathbb{R}^n$ we have that $f(x^*) = 0$ and $h(x^*) = 0$. We consider the following observation system

$$\dot{x} = f(x) \tag{14}$$

$$y = h(x) \ , \tag{15}$$

where y is interpreted as observed function.

Definition 1. *Observation system (14)-(15) is called* locally observable *near equilibrium x^*, over a given time interval $[0,T]$, if there exists $\varepsilon > 0$, such that for any two different solutions x and $\overline{x}$ of system (14) with $|x(t) - x^*| < \varepsilon$ and $|\overline{x}(t) - x^*| < \varepsilon$ $(t \in [0,T])$, the observed functions $h \circ x$ and $h \circ \overline{x}$ are different. ($\circ$ denotes the composition of functions. For brevity, the reference to $[0,T]$ will be suppressed).*

For the formulation of a sufficient condition for local observability consider the linearization of the observation system (14)-(15), consisting in the calculation of the Jacobians

$$A := f'(x^*) \text{ and } C := h'(x^*) \ .$$

Theorem 1. *([8]). Suppose that*

$$rank[C|CA|CA^2|\ldots|CA^{n-1}]^T = n\ .$$

Then system (14)-(15) is locally observable near equilibrium x^*.

Now, we recall the construction of an observer system will be based on [13].

Definition 2. *Given a continuously differentiable function $G : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, system*

$$\dot{z} = G(z, y) \tag{16}$$

is called a local asymptotic (respectively, exponential) observer for observation system (14)-(15) if the composite system (14)-(15),(16) satisfies the following two requirements.

i) If $x(0) = z(0)$, then $x(t) = z(t)$, for all $t \geq 0$.
ii) There exists a neighbourhood V of the equilibrium x^* of $\mathbb{R}^n$ such that for all $x(0) = z(0) \in V$, the estimation error $z(t) - x(t)$ decays asymptotically (respectively, exponentially) to zero.

Theorem 2. *([13]). Suppose that x^* is a Lyapunov stable equilibrium of (14), and that there exists a matrix K such that matrix $A - KC$ is Hurwitz (i.e. its eigenvalues have negative real parts), where $A = f'(x^*)$ and $C = h'(x^*)$. Then the dynamic system defined by*

$$\dot{z} = f(z) + K[y - h(z)]$$

is a local exponential observer for observation system (14)-(15).

Now, for the estimation of a change in the dynamical parameters of an ecosystem, we recall that Sundarapandian in [13] also considered the possibility of an "input generator" determined by an external system called *exosystem* $w' = s(w)$, in terms of which we can form a composite (nonlinear) system of the form

$$\begin{aligned} \dot{x} &= F(x, u(w)) \\ \dot{w} &= s(w) \\ y &= h(x)\ , \end{aligned} \tag{17}$$

where $F : \mathbb{R}^n \times \mathbb{R}^k \to \mathbb{R}^n$, $s : \mathbb{R}^k \to \mathbb{R}^k$ are continuously differentiable and $F(x^*, 0) = 0$, $u(w^*) = 0$, $s(w^*) = 0$. Here $u(w)$ is interpreted as a time-dependent vector of system parameters of the original system (14), corresponding to right-hand side f. For the construction of an observer for the composite system we can apply the following

Theorem 3. *([13]). Suppose that observation system (17) is Lyapunov stable at equilibrium. If there exists a matrix K such that matrix $A - KC$ is Hurwitz, where $A = F'(x^*, w^*)$ and $C = h'(x^*)$. Then dynamic system defined by*

$$\dot{z} = F(z, u(w)) + K[y - h(z)]$$

is a local exponential observer for observation system (17).

Limitations of Using Mass Action Kinetics Method in Modeling Biochemical Systems: Illustration for a Second Order Reaction

Cigdem Sayikli and Elife Zerrin Bagci

Biology Department, Faculty of Arts and Sciences, Namik Kemal University, Tekirdag, Turkey
ebagci@nku.edu.tr

Abstract. Mathematical models for solving the nonlinear interactions in biological systems are a major component of computational systems biology. Deterministic method that is generally employed in these models assumes that the number of molecules in the well mixed reaction medium is large. When these assumptions are not valid, the system is stochastic. In deterministic methods, mass action kinetics terms are usually included. Herein, we illustrate that the results of deterministic mass action kinetics methods do not necessarily represent the average behavior of stochastic systems with a simple example of second order reaction.

Keywords: computational systems biology, mass action kinetics, deterministic methods, illustration of stochasticity concept, stochasticity in biochemical systems.

1 Introduction

Biology is a science that studies life, by utilizing scientific observations and experimenting as well as engineering of complex systems. Systems biology that is a newly emerging field in biology, has two aspects, computational and experimental and it aims to study and analyze complex biological systems. A common method in modeling biochemical networks is nonlinear dynamics. Many mathematical models of signaling networks are based on ordinary differential equations (ODEs), generally using mass action kinetics to model biochemical reactions. These models compose biologically specific formulations, such as enzyme kinetics description and positive and negative feedback regulations [1], [2].

The use of ODEs is acceptable if the numbers of molecules of the species of all of the modeled species are large enough that their concentrations are continuous variables. Otherwise, the numbers of individual molecules should be increased or decreased in the simulations. Additionally, ODE based models have the assumption that the systems are well mixed and the species are distributed homogeneously in space. The systems with large numbers of molecules that are mixed well and distributed homogenously are deterministic with respect to their number evolution under chemical reactions (See Fig. 1). On the other hand, the systems that do not display these

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 521–526, 2011.

properties which have a known set of numbers of species evolves to several possible systems with different sets of numbers of species, are stochastic. Cellular systems are such systems because of their small volume and low number of molecules. Biological systems usually display complex spatial structure and they have low numbers of molecules and low diffusion rates which cause stochastic differences in the molecular concentrations. These differences are combined with nonlinear amplification observed in biological systems that enhances the effects of noise.

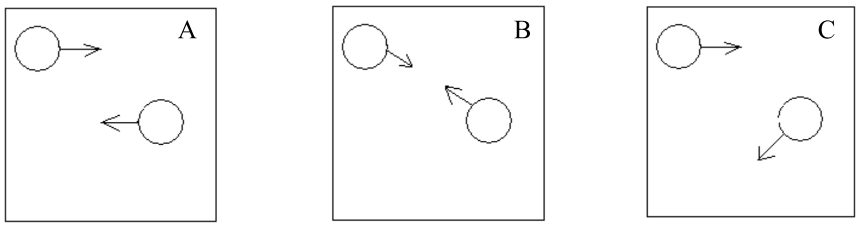


Fig. 1. Illustration for stochasticity, the different collision times for two molecules in a reaction box. (A) The case when there is no possibility of colliding (B) when the collision time is shortest (C) the molecules direction of velocities are random and represent the well mixed reaction medium which is assumed in deterministic mass action kinetics methods.

The combination of stochastic simulation with spatial heterogeneity was used to show the dynamic variability and sensitivity of bacterial chemotaxis obtained through adaptive receptor clustering [3]. On the other hand, ODE models of bacterial chemotaxis predicted robust adaptation of the response [4]. Another problem with deterministic mass action kinetics methods is the possible misconception that the average behavior of stochastic systems can be represented by deterministic models; however this is not always valid as will be illustrated by a second order reaction in the following sections. This example can be useful in the introduction of the stochasticity concept to the undergraduate/graduate students who had taken chemical kinetics courses.

2 Deterministic Mass Action Kinetics versus Stochastic Kinetics

Deterministic mass action kinetics methods assume that a known set of numbers of species in a reaction medium lead to a unique set of numbers of species at any given time. However, the chemical system is called stochastic if a known set of numbers of species in a reaction medium leads to multiple possible sets of numbers of species that occur with their corresponding probabilities.

For a second order reaction,

$$A + A \xrightarrow{k} B + C\,. \tag{1}$$

We can write the rate of formation of products, B and C, using mass action kinetics for this reaction as:

$$dN_B/dt = dN_C/dt = kN_A^2 \, . \tag{2}$$

The time evolution of number of B and C molecules can be calculated analytically or numerically using this equation if the initial number of A, B and C molecules and the value of the rate constant (k) are known.

Our aim is to evaluate the extent of the validity of this method when the number of molecules of species is low and the reaction medium is not well mixed. We note that these conditions are encountered frequently in biochemical systems. The unique species number trajectory indicates that at any given time, the reaction time is unique. However, the reaction times can differ in between the two extremes illustrated in Fig. 1A and 1B. Hence, the outcome of reactions in a reaction that is not well mixed is stochastic. Deterministic method may be used to evaluate the average behavior of a stochastic reaction system however; the mass action kinetics formula has limitations. In the following section, we evaluate the validity of using mass action kinetics formula for the bimolecular reaction mechanism presented below.

3 Limitations of Using Mass Action Kinetics Term for a Second Order Reaction

Two molecules of A diffuse towards each other within a distance of interaction, then they dimerize by the formation of a covalent bond between them. These steps are succeeded by the breaking of another covalent bond in the dimer resulting in two products, B and C. We assume that each step is elementary.

$$A + A \underset{k_D'}{\overset{k_D}{\leftrightarrow}} A.A \underset{k_B'}{\overset{k_B}{\leftrightarrow}} A_2 \xrightarrow{k_S} B + C \tag{3}$$

where A.A are pairs of A molecules with centers within an interaction distance of R. The first elementary step (D) is diffusion of A molecules within a distance of R, the second elementary step (B) is the formation of the covalent bond (hence, dimerization), and the third elementary step (S) is breaking of the second covalent bond (See Fig. 2).

3.1 Diffusion Limited Case

We assume that the diffusion step is the slowest and hence is the rate limiting step:

$$A + A \underset{k_D'}{\overset{k_D\text{,slow}}{\leftrightarrow}} A.A \underset{k_B'}{\overset{k_B}{\leftrightarrow}} A_2 \xrightarrow{k_S} B + C \tag{4}$$

then,

$$dN_B/dt = dN_C/dt = k_D N_A^2 \, . \tag{5}$$

Here, N_A is the number of molecules of A and k_D is the rate constant for diffusion of A molecules.

The rate of formation of B scales with the number of pairs of A, $N_A(N_A-1)/2$, however, mass action kinetics assume that it is proportional to N_A^2. Hence, the limitation of mass-action kinetics formula for this reaction mechanism is the magnitude of the number of molecules. In the order of number of moles (Avogadro's number, $6.02x10^{23}$) the term in mass-action kinetics formula (N_A^2) is very close to the more precise term ($N_A(N_A-1)$). On the other hand, for a reaction medium with 100 numbers of molecules of A, the error is 1%. In biochemical systems, the number of molecules is many orders of magnitude smaller than Avogadro's number which is usually in the orders of magnitude of 1, 10, 100 and 1000.

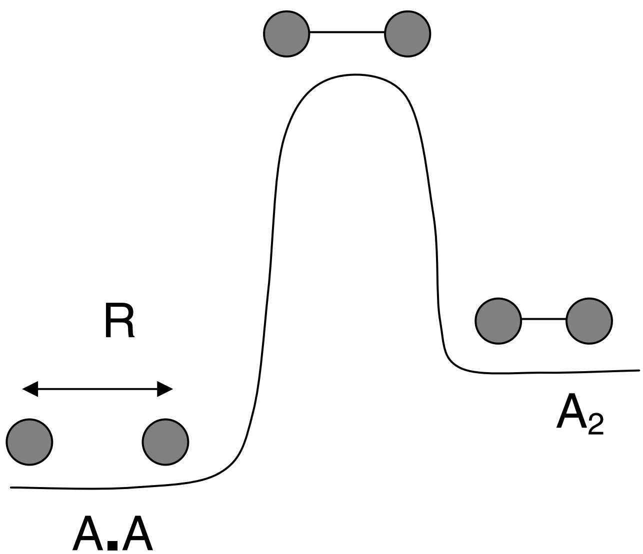


Fig. 2. Reaction coordinate diagram for the elementary bond formation step. The two A molecules diffuse towards each other within a distance of R_C, then dimerize (step B).

3.2 Bond Breaking Step Is Rate Limiting

We assume that

(i) The last step is the slowest and hence is the rate limiting step.
(ii) The intermediate step is very fast so that A_2 is in pseudo steady-state

$$A + A \underset{k_D'}{\overset{k_D}{\leftrightarrow}} A.A \underset{k_B'}{\overset{k_B}{\leftrightarrow}} A_2 \xrightarrow{k_{S,\,slow}} B + C\,. \tag{6}$$

The assumption (i):

$$dN_B/dt = k_S N_{A2}\,. \tag{7}$$

The assumption (ii):

$$dN_{A2}/dt = k_B N_{A.A} - k_B' N_{A2} - k_S N_{A2} = 0\,. \tag{8}$$

Assuming k_S is small (assumption (i)) hence ignoring the last term in (8) and rearranging (8),

$$N_{A2} = K_B N_{A.A} \qquad \text{where} \quad K_B = k_B/k_B'\,. \tag{9}$$

Combining (7) and (9),

$$dN_B/dt = k_S K_B N_{A.A} = k' N_{A.A}\,. \tag{10}$$

We illustrate the pairs of A molecules within a distance R, $N_{A.A}$, in a reaction medium with volume V and N_A molecules in Fig. 3 and calculate $N_{A.A}$ as follows:

Number of A molecules in V is N_A and if we assume that A molecules are dispersed homogenously (well mixed reaction medium), number of A molecules in the sphere in Fig. 3 is obtained as

$$N_A(R) = \frac{N_A}{V}\frac{4\pi R_c^{\,3}}{3}\,. \tag{11}$$

The number of interacting molecules with the central molecule is $N_A(R)$-1. Therefore we can count all interacting pairs by multiplying this by N_A but then by dividing by 2 because of counting the pairs twice. Therefore $N_{A.A}$ is:

$$N_{A.A} = N_A[N_A(R)-1]\,/\,2\,. \tag{12}$$

$N_{A.A}$ becomes proportional to N_A^2, therefore using mass action kinetics for this reaction is valid, only when

$$N_A(R) = \frac{N_A}{V}\frac{4\pi R_c^{\,3}}{3} >> 1\,. \tag{13}$$

Therefore, in the case of bond breaking being the rate limiting step, several assumptions are done in using mass action kinetics term.

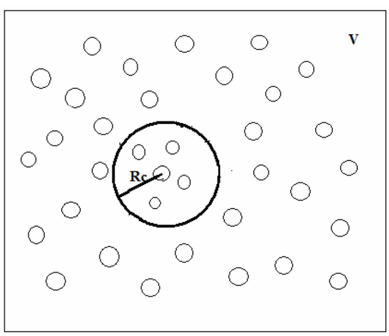


Fig. 3. The two dimensional cross-section of a reaction medium of volume V. The molecule in the middle of the sphere is interacting with molecules within a distance of R_C.

4 Conclusion

Conventionally, only mass action kinetics concept is conveyed in physics/chemistry/biology undergraduate education. This concept also has been employed generally in the newly emerging field, systems biology. However, mass action kinetics methods are deterministic as opposed to the stochastic nature of biochemical systems. Moreover, it is illustrated here that the results of mass action kinetics methods do not necessarily correspond to the mean of stochastic results. Mass action kinetics methods are much more intelligible than stochastic methods so their introduction to undergraduate students firstly can be retained. Herein, we provide a simple example that can be useful in the introduction of the stochasticity concept to the undergraduate/graduate students who had taken chemical kinetics courses and in the illustration of the shortcomings of using mass action kinetics terms in modeling systems that are not well mixed with low number of molecules.

Acknowledgment. We thank Metin Arik for useful discussions.

References

1. Ozbudak, E.M., Thattai, M., Lim, H.N., Shraiman, B.I., van Oudenaarden, A.: Multistability in the Lactose Utilization Network of Escherichia coli. Nature 427, 737–740 (2004)
2. Hasty, J., McMillan, D., Isaacs, F., Collins, J.J.: Computational Studies of Gene Regulatory Networks: In Numero Molecular Biology. Nature Rev. Genetics 2, 268–279 (2001)
3. Bray, D., Levin, M.D., Morton-Firth, C.J.: Receptor Clustering as a Cellular Mechanism to Control Sensitivity. Nature 393, 85–88 (1998)
4. Barkai, N., Liebler, S.: Robustness in Simple Biochemical Networks. Nature 387, 913–917 (1997)

The Impact of Isolation of Identified Active Tuberculosis Cases on the Number of Latently Infected Individuals

Schehrazad Selmane

Faculty of Mathematics
University of Sciences and Technology Houari Boumediene, Algiers, Algeria
schehrazad.selmane@gmail.com

Abstract. Isolation, quarantine, disinfection, inoculation, education have been the five important preventive control measures used to control an epidemic. Isolation, which is aimed at restricting the spread to susceptibles by restricting the movements of infectious cases, tops the list. Identifying and isolating patients with active tuberculosis could be effective as control measure. In order to minimize the transmission of the disease and to break the transmission chain of the Mycobacterium tuberculosis and thus the sterilization of the source of infection, we consider an optimal control strategy associated with isolation of infectious individuals who spread the disease. The existence of an optimal control for an objective functional that takes into account both the number of infectious individuals and the cost of isolation strategy, the characterization of the optimal control, and the uniqueness of the optimality system are proved. The optimality system is solved numerically using the Forward-Backward Sweep method. The numerical results showed that isolation strategy will considerably reduce the number of latently infected individuals.

Keywords: Deterministic Model, Forward-Backward Sweep Method, Isolation, Optimal control, Pontryagin's Maximum Principle, Tuberculosis.

1 Introduction

Hundred and twenty nine years after the identification by Robert Koch in 1882 of the Mycobacterium tuberculosis (MTB), pathogen of tuberculosis (TB), the disease is still and will remain a problem of public health world for several decades during the $3rd$ millennium in spite of the fact that is a preventable and curable disease (Isoniazid is used to prevent individuals latently infected with MTB from developing disease, and regimens consisting of multiple drugs are used in curing active cases with high success; between 1995 and 2009, a total of 41 million TB patients were successfully treated under the direct observation therapy strategy ($DOTS$), and up to 6 million lives were saved). In 2009, there were an estimated 9.4 million incident cases of TB globally. The absolute number of cases continues

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 527–536, 2011.

to increase slightly from year to year, as slow reductions in incidence rates per capita continue to be outweighed by increases in population. Approximately 1.7 million people died of TB in 2009 with an estimated 1.3 million deaths among HIV-negative and 0.38 million deaths among HIV-positive people. There were 5.8 million notified cases of TB in 2009, equivalent to a case detection rate of 63%. Most of the estimated number of cases in 2009 occurred in Asia (55%) and Africa (30%); 3 smaller proportions of cases occurred in the Eastern Mediterranean Region (7%), the European Region (4%) and the Region of the Americas (3%). An estimated $11-13\%$ of incident cases were HIV-positive; the African Region accounted for approximately 80% of these cases [6].

According to the World Health Organization (WHO) a third of the world's population is thought to harbor MTB; forming thus a huge latent MTB global reservoir. An intervention that target latently infected with TB ($LTBI$) individuals could be effective as control measure. However, providing treatment for a large fraction of the population is costly, besides which the identification of $LTBI$ individuals is, in a way, not feasible. This renders the prospect of ever eliminating MTB from the human race almost impossible and therefore the need to settle for global TB control rather than eradication. In order to minimize the transmission of the disease and to break the transmission chain of the MTB and thus the sterilization of the source of infection, we will develop an epidemiological TB model to assess the impact of isolation of identified active TB on the number of $LTBI$ individuals.

The paper is organized as follows : in Section 2 a $SLIT$ deterministic TB model, where a control term representing isolation of active TB individuals is incorporated, is presented and an objective functional that takes into account both the number of infectious individuals and the cost of isolation strategy is introduced. The existence of an optimal control, its characterization, the uniqueness of the optimality system are also investigated in this Section. Section 3 includes some numerical studies of the proposed optimal control and discusses the obtained results.

2 Model Formulation and Analysis of Optimal Control

TB is an infectious disease caused by bacteria called MTB, which is transmitted quasi exclusively by air. The infecting droplets are produced in the form of aerosol by the contagious patient at the time of cough, speech or sneezes. These droplets remain in suspension in the ambient air; ninety percent of them are inactivated as soon as their emission and only a fraction of 1% survive for approximately two hours. The inhalation into the lungs of some bacteria suspended in the air constitutes, in practice, the only mode of contamination. The person becomes infected by breathing in the bacteria. The immune system is sometimes able to kill TB bacteria. If not, either the bacteria remain alive but inactive in the body and the person contracts a TB infection, or they become active and begin to multiply in the body and cause TB disease. Latently Infected individuals who did not progress to TB disease may remain infected, non-infectious, for their lifetime unless endogenous reactivation or exogenous re-infection occurs [4].

The model we consider monitors the dynamics of four sub-populations, namely susceptible $S(t)$, latently infected $L(t)$, infectious $I(t)$, and treated $T(t)$ individuals; for a review of mathematical models of TB see for instance [2]. We assume that the population is homogeneous mixed, and all people are equally likely to be infected by an infectious individual in a case of adequate contact. We assume that individuals are recruited into the population either by birth or immigration at rate $\Lambda = \mu N$ where the natural death rate, μ, in each class is assumed to be positive. The disease induced death rate is denoted by d. Susceptible individuals can be infected only through contact with individuals having smear positive pulmonary TB disease at rate $\lambda = \beta I/N$; a proportion p degenerates into infectious class, whereas the remaining proportion degenerates into latent infection class. The latently infected individuals progress to active TB at low risk and slowly to infectious class either, by endogenous reactivation at rate α, or, by exogenous re-infection at rate $\sigma_L \beta I$, where σ_L is a factor reducing the risk of infection as a result of acquired immunity to a previous infection. All detected infectious individuals receive treatment at rate τ. A fraction of those whose treatment were unsuccessful relapse back into the infectious class at rate r; they receive a treatment of second line [4]. The patients whose cases are not detected either die or self cure at rate e. Successfully treated individuals acquire some immunity not fully which reduces the risk of re-infection. They can return to the latent infection class only by exogenous re-infection at rate $\sigma_T \beta I$ where σ_T is the factor reducing the risk of re-infection, as a result of acquired immunity to a previous infection and treatment. An intervention to keep TB under control, in addition to the prompt detection and treatment of active TB cases, may consist on identification and isolation of smear positive pulmonary TB individuals till they become smear negative, and isolation of treatment failure cases and drugs resistant cases. A control function, $u(t)$, is incorporated into the model, it represents the effort on isolation of infectious, treatment failure, and drugs resistant TB patients in order to break the transmission of the disease. The model is schematically illustrated in Fig. 1 and the interactions of the four compartments are specified by the following system

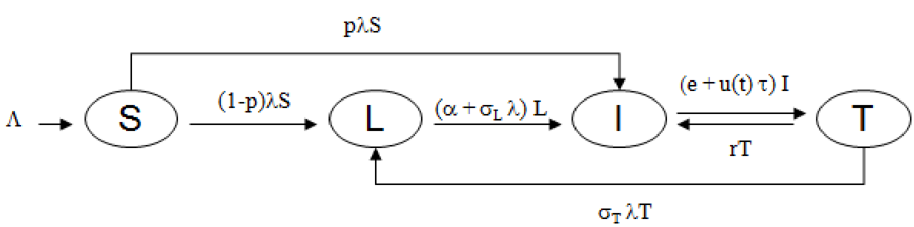


Fig. 1. Flows between the compartments of the model

$$\begin{cases} \dot{S} = \Lambda - \beta \frac{I}{N} S - \mu S \\ \dot{L} = (1-p)\beta \frac{I}{N} S + \sigma_T \beta \frac{I}{N} T - \left(\sigma_L \beta \frac{I}{N} + \alpha + \mu\right) L \\ \dot{I} = p\beta \frac{I}{N} S + \left(\sigma_L \beta \frac{I}{N} + \alpha\right) L + rT - \left(e + u(t)\tau + d + \mu\right) I \\ \dot{T} = -\sigma_T \beta \frac{I}{N} T + \left(e + u(t)\tau\right) I - (r+\mu) T \end{cases} \tag{I}$$

where $S(0)$, $L(0)$, $I(0)$, $T(0)$ are given and the control function, u, is bounded, Lebesgue integrable function. Our control problem involves the numbers of infectious individuals, and the cost of implementing isolation strategy. This led us to define the objective functional to be minimized as

$$J(u) = \int_0^{T_f} \left(A\ I(t) + \frac{B}{2}\ u^2(t)\right) dt \tag{II}$$

subject to the state system (I). The coefficients A and B are balancing cost factors due to size and importance of the two parts of the integrand in the objective functional (II). We require $B > 0$ because it is a minimization problem. The objective is to find an optimal control u^* and the corresponding path such that

$$J(u^*) = \min_{\Omega} J(u) \tag{1}$$

where $\Omega = \left\{u(t) \in L^1(0, T_f) : a \leq u(t) \leq b,\ 0 \leq t \leq T_f\right\}$ and a, b are fixed positive constants.

2.1 Method for Finding Optimal Solution

After formulating the model and given the objective functional appropriate to our goal, we have to prove the existence of an optimal control, its characterization, the uniqueness of the optimality system, and finally solve it numerically. In our analysis, we treat the case of constant population; the nonconstant population case can be done using the same techniques. We assume $\Lambda = \mu N$ and $d = 0$; thus the total population N is constant.

We set $X = X(t) = \left(S(t), L(t), I(t), T(t)\right)^t$, $X_0 = X(0)$, and $\lambda = \lambda(t) = \left(\lambda_1(t), \lambda_2(t), \lambda_3(t), \lambda_4(t)\right)^t$.

Existence. The control space Ω is convex and closed by definition and the state system with respect to the state variables is uniformly Lipschitz continuous, and the integrand, $A\ I(t) + \frac{B}{2}\ u^2(t)$, in the objective functional (II) is convex on the control $u(t)$, and there exists a constant $\nu > 1$ and positive numbers ξ_1 and ξ_2 such that $J(u) \geq \xi_2 + \xi_1 \left\|u\right\|^2$. Therefore, and according to theorem 2.2 [3], there exists an optimal control function u^* satisfying (1) .

Characterization. Pontryagin proved that the necessary condition to solve an optimal control problem is to choose a control so as to minimize pointwise the Hamiltonian, $\mathcal{H}$,

$$\mathcal{H} = A\ I(t) + \frac{B}{2}\ u^2(t) + \sum_{i=1}^{4} \lambda_i f_i \tag{2}$$

with respect to u (see [3]), where f_i is the right hand side of the differential equation of the $i-th$ state variable and where λ_i $(1 \leq i \leq 4)$ are so-called adjoint variables. That is

$$\begin{aligned}
\mathcal{H} = A\ I(t) + \frac{B}{2}\ u^2(t) + \lambda_1 \left[\Lambda - \beta\frac{I}{N}S - \mu S\right] \\
+\lambda_2 \left[(1-p)\beta\frac{I}{N}S + \sigma_T\beta\frac{I}{N}T - \left(\sigma_L\beta\frac{I}{N} + \alpha + \mu\right) L\right] \\
+\lambda_3 \left[p\beta\frac{I}{N}S + \left(\sigma_L\beta\frac{I}{N} + \alpha\right) L + rT - (e + u(t)\tau + \mu)\, I\right] \\
+\lambda_4 \left[-\sigma_T\beta\frac{I}{N}T + (e + u(t)\tau)\, I - (r+\mu)\, T\right]
\end{aligned}$$

Theorem 1. *Given an optimal control variable u^* and solutions of the corresponding state system for the optimal control problem (I) and (II), there exist adjoint variables λ_i for $i = 1, ..., 4$ that satisfy*

$$\begin{cases}
\frac{d\lambda_1}{dt} = \lambda_1 \left(\beta\frac{I}{N} + \mu\right) - \lambda_2(1-p)\beta\frac{I}{N} - \lambda_3 p\beta\frac{I}{N} \\
\frac{d\lambda_2}{dt} = \lambda_2 \left(\sigma_L\beta\frac{I}{N} + \alpha + \mu\right) - \lambda_3 \left(\sigma_L\beta\frac{I}{N} + \alpha\right) \\
\frac{d\lambda_3}{dt} = -A\ + \lambda_1\beta\frac{S}{N} - \lambda_2 \left[(1-p)\beta\frac{S}{N} + \sigma_T\beta\frac{T}{N} - \sigma_L\beta\frac{L}{N}\right] \\
\qquad -\lambda_3 \left[p\beta\frac{S}{N} + \sigma_L\beta\frac{L}{N} - (e + u(t)\tau + \mu)\right] \\
\qquad -\lambda_4 \left[-\sigma_T\beta\frac{T}{N} + (e + u(t)\tau)\right] \\
\frac{d\lambda_4}{dt} = -\lambda_2\sigma_T\beta\frac{I}{N} - \lambda_3 r + \lambda_4 \left[\sigma_T\beta\frac{I}{N} + (r+\mu)\right]
\end{cases} \tag{3}$$

with transversality conditions

$$\lambda_i(T_f) = 0, \qquad i = 1, ..., 4 \tag{4}$$

Furthermore the following characterization (called the characterization of the optimal control) holds

$$u^*(t) = \min\left(\max\left(a, \frac{1}{B}\left(\lambda_3 - \lambda_4\right)\tau I\right), b\right). \tag{5}$$

Proof. To determine the adjoint equations and the transversality conditions, we use the Hamiltonian (2). Applying Pontryagin's Maximum principle [3], the differential equations governing the adjoint variables are obtained by differentiating the Hamiltonian function with respect to S, L, I and T

$$\begin{aligned}
\frac{d\lambda_1}{dt} &= -\frac{\partial\mathcal{H}}{\partial S}, \qquad \lambda_1(T_f) = 0, \\
\frac{d\lambda_2}{dt} &= -\frac{\partial\mathcal{H}}{\partial L}, \qquad \lambda_2(T_f) = 0, \\
\frac{d\lambda_3}{dt} &= -\frac{\partial\mathcal{H}}{\partial I}, \qquad \lambda_3(T_f) = 0, \\
\frac{d\lambda_4}{dt} &= -\frac{\partial\mathcal{H}}{\partial T}, \qquad \lambda_4(T_f) = 0,
\end{aligned}$$

evaluated at the optimal control and corresponding states, which result in the stated adjoint system (3) and transversality conditions (4).

The characterization (5) of the optimal control u^* is derived from the optimality condition $\frac{\partial \mathcal{H}}{\partial u} = 0$, namely,

$$u(t) = \frac{1}{B}\left(\lambda_3 - \lambda_4\right)\tau I$$

and by taking into account the bound on u^*.

Uniqueness. Incorporating the representation of the optimal isolation control, we obtain the state system coupled with the adjoint system which characterizes the optimal control, and is called the optimality system:

$$\begin{cases} \dot{S} = & \Lambda - \beta\frac{I}{N}S - \mu S \\ \dot{L} = & (1-p)\beta\frac{I}{N}S + \sigma_T\beta\frac{I}{N}T - \left(\sigma_L\beta\frac{I}{N} + \alpha + \mu\right)L \\ \dot{I} = & p\beta\frac{I}{N}S + \left(\sigma_L\beta\frac{I}{N} + \alpha\right)L + rT - \\ & \left(e + \min\left(\max\left(a, \frac{1}{B}\left(\lambda_3 - \lambda_4\right)\tau I\right), b\right)\tau + \mu\right)I \\ \dot{T} = & -\sigma_T\beta\frac{I}{N}T + \left(e + \min\left(\max\left(a, \frac{1}{B}\left(\lambda_3 - \lambda_4\right)\tau I\right), b\right)\tau\right)I - (r+\mu)T \\ & X(0) = X_0 \\ \dot{\lambda}_1 = \lambda_1\left(\beta\frac{I}{N} + \mu\right) - \lambda_2(1-p)\beta\frac{I}{N} - \lambda_3 p\beta\frac{I}{N} \\ \dot{\lambda}_2 = \lambda_2\left(\sigma_L\beta\frac{I}{N} + \alpha + \mu\right) - \lambda_3\left(\sigma_L\beta\frac{I}{N} + \alpha\right) \\ \dot{\lambda}_3 = -A\ + \lambda_1\beta\frac{S}{N} - \lambda_2\left[(1-p)\beta\frac{S}{N} + \sigma_T\beta\frac{T}{N} - \sigma_L\beta\frac{L}{N}\right] \\ \quad -\lambda_3\left[p\beta\frac{S}{N} + \sigma_L\beta\frac{L}{N} - \left(e + \min\left(\max\left(a, \frac{1}{B}\left(\lambda_3 - \lambda_4\right)\tau I\right), b\right)\tau + \mu\right)\right] \\ \quad -\lambda_4\left[-\sigma_T\beta\frac{T}{N} + \left(e + \min\left(\max\left(a, \frac{1}{B}\left(\lambda_3 - \lambda_4\right)\tau I\right), b\right)\tau\right)\right] \\ \dot{\lambda}_4 = -\lambda_2\sigma_T\beta\frac{I}{N} - \lambda_3 r + \lambda_4\left[\sigma_T\beta\frac{I}{N} + (r+\mu)\right] \\ \quad \lambda(T_f) = 0. \end{cases} \tag{III}$$

The uniqueness of the optimal control follows from the uniqueness of the optimality system (III). In order to guarantee the uniqueness of the optimality system there is a restriction on the length of the time interval due to opposite time orientations; the state system moves forward in time and has initial values whereas the adjoint system moves backward in time and has final values. Thus the uniqueness of the optimal control pair is guarantee for small T_f.

3 Numerical Illustrations and Conclusions

3.1 Numerical Algorithm

The existence and uniqueness have been established, the Two Point Boundary Value Problem expressed by the optimality system (III), is solved numerically using the Forward-Backward Sweep method [3] sketched in the sequel.

1. Store an initial guess for the control u over the interval $[a, b]$ ($u \equiv 0$ is almost always sufficient).

2. Using the initial condition X_0 for the state X and the stored value for u, solve X forward in time according to its differential equation in the optimality system (we used the forward Runge-Kutta fourth order procedure).
3. Using the transversality condition $\lambda(T_f) = 0$ and the stored values for u and X, solve λ backward in time according to its differential equation in the optimality system (we used the backward Runge-Kutta fourth order procedure).
4. Update the control by entering the new X and λ values into the characterization of u.
5. Check the convergence. If the values of the variables in this iteration and the last iteration are negligibly small, output the current values as solutions. Otherwise, return to step 2.

3.2 Simulations and Conclusion

The simulations were carried out using the following biological feasible parameter values. Since the model considers human populations, all the model parameters and variables are assumed to be non-negative. The birth/non-TB death rate, μ, is $1/70$ year^{-1}, corresponding to the life expectancy 70 years. The parameters modelling TB were taken from the literature. Following estimates of Vynnycky and Fine [5], we took $p = 0.11$ per year and $\alpha = 0.000113$ per year. For the rate of self-cure we used $e = 0.25$ per year [1], and the recovery rate from TB with treatment, $\tau = \gamma\theta \simeq 0.63$, where the probability of successful treatment for detected active TB cases θ is approximately 90%, and the detection rate of active TB cases γ is approximately 70% [1]. As regards the rate of relapse from chemotherapeutic cure we took $r = 0.005$ per year and the factors reducing the risk of infection σ_L and σ_T were considered variable. For a fixed value of the population size N; we set the initial values $I_0 = N/100$; $L_0 = I_0/p$; $R_0 = \tau I_0$; $S_0 = N - L_0 - I_0 - R_0$.

In practice it is very hard to acquire the appropriate weights; it needs a lot of work on data mining, analyzing, and fitting. The weight factors in our simulations are of theoretical interest to illustrate the optimal isolation strategy; we considered several values for A and B. The cost associated with u will include the cost of holding the patients in hospital and treatment. We chose the upper bound, b, equal to 0.95 and the lower bound, a, equal to 0.05.

Fig. 2 shows the outcome of the model for latently infected and infectious individuals when the optimal isolation strategy is applied (solid curve) and when only treatment is applied (dotted curve) for $N = 6000$, $\beta = 11$, $\sigma_L = 0.25$ and $\sigma_T = \frac{1}{4}\sigma_L$. We observe that the total number of latently infectious and infectious individuals at final time $T_f = 6$ with optimal control are respectively $L_6 = 1067$ and $I_6 = 1249$ and $L_6 = 1449$ and $I_6 = 1590$ without control, thus the total number of latently infected and infectious prevented after 6 years of application of the isolation strategy is 382, 341 respectively.

In Fig. 3, the control is plotted as a function of time for $N = 6000$, $\sigma_T = \sigma_L = 0.25$, $A = 10$, $B = 1$, and for different values of the effective contact rate β; we used $\beta = 7$ for low, $\beta = 10$ for mild, and $\beta = 13$ for severe epidemic.

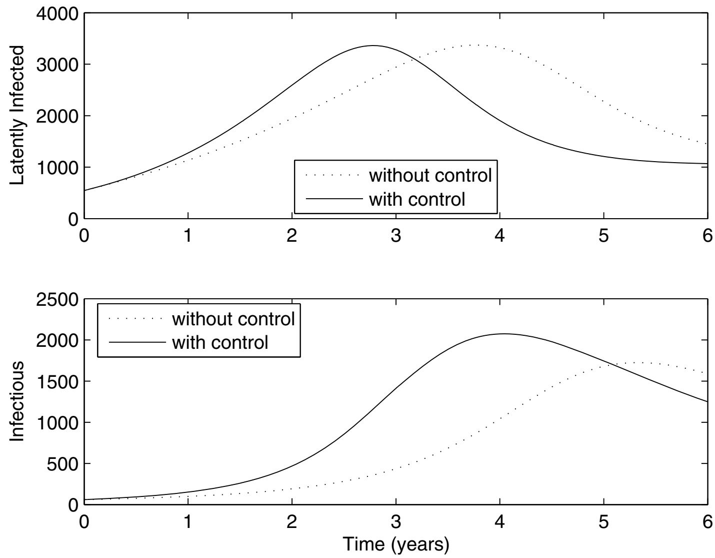


Fig. 2. The number of latently infected and infectious individuals are plotted as a function of time for the case of $A = 1$ and $B = 1/4$ in the top and bottom frame respectively

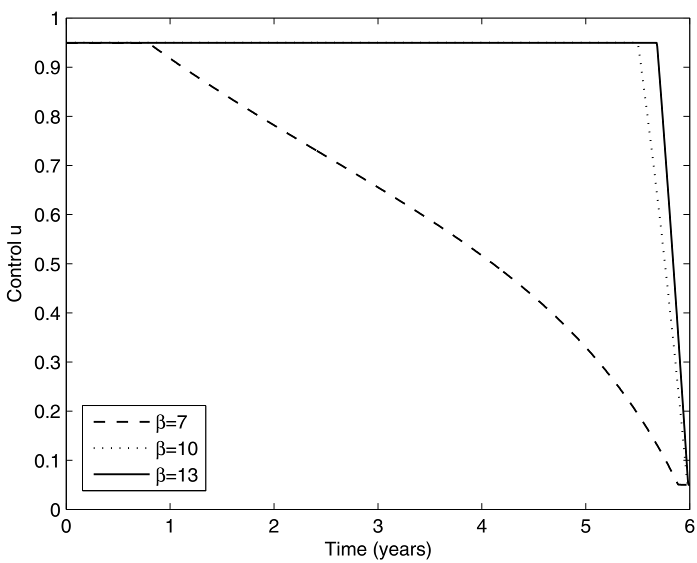


Fig. 3. Profile of the control function u for three different values of β, 7, 11 and 13

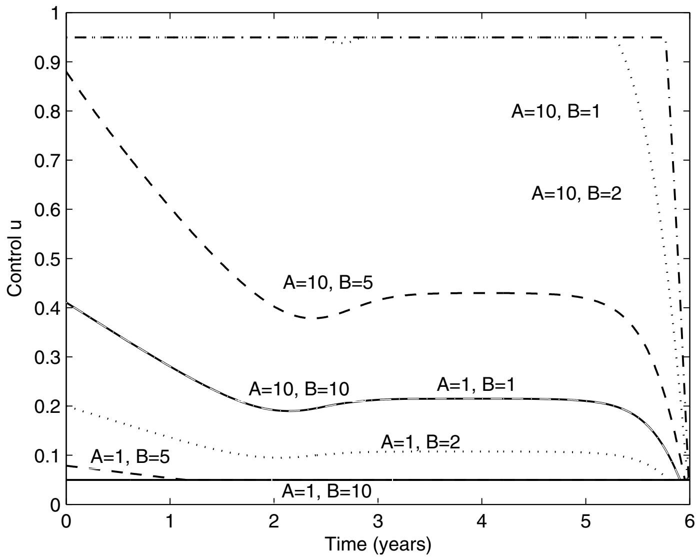


Fig. 4. Plot of the control function u as a function of time for the different values of A and B

For each of the fixed value of β, the control u remains the same as the population size increases.

For a fixed value of β, the control u remains almost the same for any fixed value of A and B such that A=B. Note that with A fixed, as B increases, the amount of the control u decreases as shown in Fig. 4 where the control is plotted as a function of time for several values of B ($B = 1, 2, 5, 10$) while the weight A is fixed to $A = 1$ and $A = 10$, and using the following parameters values : $N = 6000$, $\beta = 9$, $\sigma_T = \sigma_L = 0.65$.

The results show how the optimal control isolation strategy may depend on factors reducing the risk of infection σ_L and σ_T and the effective contact rate; control programs that follow this strategy can effectively reduce the number of latent TB cases.

References

1. Bacaer, N., Ouifki, R., Pretorius, C., Wood, R., Williams, B.: Modeling the joint epidemics of TB and HIV in a south african township. J. Math. Biol. 28(4), 365–382 (2008)
2. Castillo-Chavez, C.: Dynamical models of Tuberculosis and their applications. Mathematical Biosciences and Engineering 1, 361–404 (2004)
3. Lenhart, S., Workman, J.T.: Optimal control applied to biological models. In: Mathematical and Computational Biology Series. Chapman&Hall/CRC, London (2007)

4. Manuel de la lutte antituberculeuse à l'usage des personnels médicaux – Programme National de Lutte Contre la Tuberculose- MSP, Direction de la prévention- INSP, édition (2001)
5. Vynnycky, E., Fine, P.E.M.: The natural history of tuberculosis: the implications of age-dependent risks of disease and the role of reinfection. Epidemiol. Infect. 119, 183–201 (1997)
6. World Health Organization (WHO), Global Tuberculosis Control. WHO report 2010. WHO/HTM/TB/2010.7

Performance Analysis of an Algorithm for Computation of Betweenness Centrality

Shivam Bhardwaj[1], Rajdeep Niyogi[1], and Alfredo Milani[2]

[1] Department of Electronics and Computer Engineering
Indian Institute of Techology Roorkee
Roorkee- 247667 India
rajdpfec@iitr.ernet.in
[2] Department of Mathematics and Computer Science
University of Perugia, via Vanvitelli 1 06100, Perugia, Italy
milani@dmi.unipg.it

Abstract. In social network analysis, graph-theoretic perceptions are used to realize and explain social experience. Centrality indices are essential in the analysis of social networks, but are costly to compute. An efficient algorithm for the computation of betweenness centrality is given by Brandes that has time complexity $O(nm + n^2 \log n)$ and $O(n + m)$ space complexity, where n, m are the number of vertices and edges in a graph, respectively. Some social network graphs are invariably huge and dense. Moreover, size of memory is rapidly increasing and the cost of memory is decreasing day by day. Under these circumstances, we investigate how the computation of centrality measures can be done efficiently when space is not very significant. In this paper, we introduce a time efficient and scalable algorithm for the accurate computation of betweenness centrality. We have made a thorough analysis of our algorithm vis-à-vis Brandes' algorithm. Experimental results show that our algorithm has a better performance with respect to time, but at the expense of using higher memory. Further performance improvement of our algorithm has been achieved by implementing it on parallel architectures.

Keywords: Social networks, betweenness centrality.

1 Introduction

A social network is an association of people drawn together by family, work, friendship, or hobby. Online social networks have gained significant popularity and are now among the most popular sites on the Web [1]. Such a network represents a social structure that can be modeled as a graph. The nodes in the graph represent individuals or organizations. The edges represent one or more specific types of interdependency, such as friendship, kinship, financial exchange, and dislike, relationships of beliefs, knowledge, or prestige. A user's links, along with the user profile, are visible to those who visit the user's account. Thus users are able to explore the social network by following user-to-user links, browsing the profile information and any contributed content of visited users. Certain sites, such as LinkedIn, only allow a user to browse

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 537–546, 2011.

other users' accounts within its neighborhood i.e. a user can only view other users that are within two hops in the social network. Whereas other sites like Orkut allow users to view any other user account in the system. For an inclusive impression of methods and applications we refer to [2].

An essential metric for the analysis of social networks is centrality indices [3] of the nodes of the graph. For analyzing data on social networks, researchers are often interested in different types of centrality, which are measures of a user's importance in a network [4]. These are designed to rank the users according to their position in the network and construed as the importance of a user embedded in a social structure. These measures give a rough indication of the social power of a node based on how well they connect the network.

With the increasing demand of the electronic data collection and its analysis, the demand for the computation of centrality indices on networks with thousands of users is also increasing. However, existing implementations are too much time expensive, which makes the computation of principally important betweenness centrality index costly for large number of nodes. As a remedy, network analysts are now suggesting simpler indices, for instance based only on linkages between the neighbors of each user [5], to at least obtain rough approximations of betweenness centrality.

In this paper, we introduce a time efficient and scalable algorithm for the accurate computation of betweenness centrality based on shortest path distance between pair of nodes. The number of nodes for which betweenness centrality can be computed is thus increased significantly and time for computation of other centrality measures based on shortest path is also reduced.

We have further improved the performance of our algorithm using parallelization with the help of *pthread* s. We have also implemented the algorithm on a cluster of computers.

2 Centrality Measures of Social Networks

Many centrality indices are based on shortest paths linking pairs of nodes or the ratio of shortest paths that a node lies on. Many social network-analytic studies rely on, at least in part, an evaluation of these indices. Different centrality measures based on shortest path are as follows:

2.1 Betweenness Centrality

Betweenness is a centrality measure of a node within a graph which represents the extent to which a node lies between the other nodes in the network. Nodes that occur on many shortest paths between other nodes have higher betweenness than those that do not. In social network point of view, the measure reflects the number of people who a person is connecting indirectly through their direct links [4]. For example in figure 1, node X is connecting node A and B through its direct link. Here, nodes A and B are connected indirectly with shortest path length two. There are two such paths out of which one is passing through node X. So in this case, betweenness of node X will be 0.5.

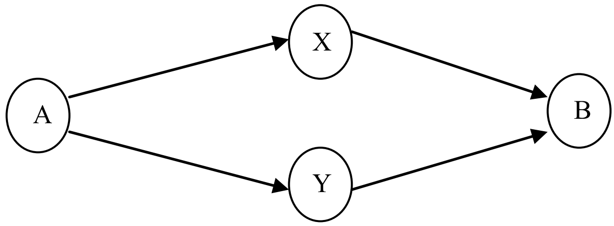


Fig. 1. Example of user graph

For a graph *G*: = (V,E) where V is set of nodes and E is set of edges, the betweenness centrality $C_B(v)$ for a node *v* is:

$$C_B(v) = \sum_{s \neq v \neq t \in V} \left(\frac{\sigma_{st}(v)}{\sigma_{st}} \right)$$

σ_{st} represents the number of shortest paths from node s to t and $\sigma_{st}(v)$ represents the number of shortest paths from s to t which passes through *v*.

2.2 Closeness Centrality

Closeness is a centrality measure of a node within a graph which represents the degree an individual is near all other individuals in a network, directly or indirectly [6]. Nodes that have shorter distances to other nodes within the graph have higher closeness. Closeness is preferred in network analysis as it gives higher values to more central nodes, and so is usually positively associated with other measures such as degree.

For a graph *G*: = (V,E) where V is set of nodes and E is set of edges, the closeness centrality $C_C(v)$ for node *v* is:

$$C_c(v) = \frac{1}{\sum_{t \in V} d_G(v,t)}$$

$d_G(v,t)$ is the shortest path length from node *v* to *t*. This shortest path length for each pair of node is present in the shortest path matrix.

2.3 Stress Centrality

Stress centrality of a node in the graph is similar to the betweenness centrality. It is calculated as the summation of number of shortest paths between all possible pair of nodes passing through the node in question.

For a graph *G*: = (V,E) where V is set of nodes and E is set of edges, the stress centrality $C_S(v)$ for node *v* is:

$$C_S(v) = \sum_{s \neq v \neq t \in V} \left(\sigma_{st}(v) \right)$$

$\sigma_{st}(v)$ represents the number of shortest path from node *s* to *t* which passes through node *v*.

2.4 Graph Centrality

Graph centrality of a node is the invert of the maximum of all shortest distances from that node to all other nodes in the network. Nodes with higher Graph centrality value have shorter distances to all other nodes in the graph.

For a graph G: = (V,E) where V is set of nodes and E is set of edges, the graph centrality $C_G(v)$ for node v is:

$$C_G(v) = \frac{1}{\max_{t \in V} d_G(v,t)}$$

$d_G(v,t)$ is the shortest path length from node v to t.

3 An Algorithm for Betweenness Centrality

In this section, we propose an algorithm for calculation of betweenness centrality of a directed graph--representing a social network.

3.1 Computation of Shortest Path Matrix

The length of a shortest path and the number of shortest paths between each pair of nodes are stored in a two dimensional matrix with each cell containing two values-shortest distance from the node with index equal to row number to the node with index equal to column number and the number of shortest paths between these two nodes.

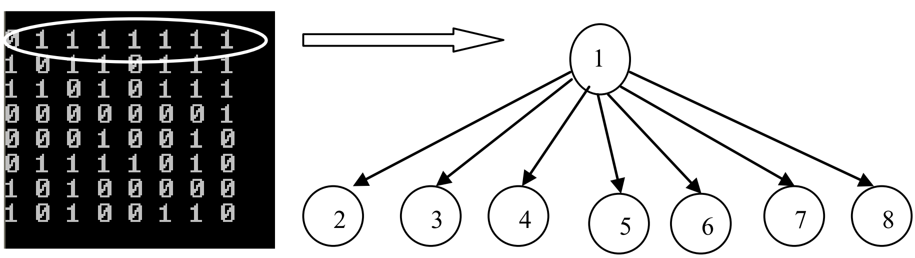


Fig. 2. Example of Random directed graph

Figure 3 shows the shortest path matrix for the graph shown in figure 2. In this figure, each row represents the shortest distance from node with index equal to the row number to other nodes in the graph. For example, the second row represents the shortest distance of all nodes from node 2. First column of second row is 1,1 which represents that node 1 is at a distance equal to 1 from node 2 and number of shortest path is 1.

Before the calculation of the shortest path matrix, it is essential to note that we don't need to know the shortest path but only the length of the shortest path between the pair of nodes. We also need to know how many such paths exist. Thus, popular shortest path algorithms like Dijkstra's algorithm cannot be applied here as it returns the shortest path between a pair of nodes and not the number of shortest paths that exists between them [7].

0,0	1,1	1,1	1,1	1,1	1,1	1,1	1,1
1,1	0,0	1,1	1,1	2,2	1,1	1,1	1,1
1,1	1,1	0,0	1,1	2,2	1,1	1,1	1,1
2,1	3,3	2,1	0,0	3,2	2,1	2,1	1,1
2,1	3,2	2,1	1,1	0,0	3,3	1,1	2,1
2,3	1,1	1,1	1,1	1,1	0,0	1,1	2,3
1,1	2,2	1,1	2,2	2,1	2,2	0,0	2,2
1,1	2,3	1,1	2,3	2,2	1,1	1,1	0,0

Fig. 3. Shortest path matrix

In order to calculate the shortest path matrix, we calculate the shortest distance from node *i* to all other nodes in the graph. So first, for each node *i*, all nodes *j* which are neighbor of *i* were explored. The shortest distance from node *i* to *j* will be 1 and shortest path count will also be 1. Now we increase the distance by one and explore all neighbors of *j* and if it is not yet visited, then the distance and number of visit with that distance are stored in the shortest path matrix. This continues till we have no unvisited node left. This is repeated for all nodes *i* in the graph. The pseudo code for the algorithm is given below where *length(i,j)* represents the length of the shortest path between node *i* and *j* and *path_count(i, j)* represents number of such paths.

```
Pseudo code for computation of Shortest Path Matrix

for all nodes i
{
   int *x;
   for all nodes j :  j ∈ neighbor of i
   ENQUEUE(Q,[j, length(i,j)])

   while Q not empty
   {
     x ← DEQUEUE(Q);
     for all nodes k : k ∈ neighbors of x  and  k ∉
     neighbors of i
     {
       if length(i, k) =  length(i,x) + 1 then
       path_count(i, k)++ ;
       else
       if length(i, k) is not yet computed then
       {
          length(i, k) ← length(i,x) + 1;
          path_count(i, k)++ ;
          ENQUEUE(Q, [k, length(i, k)]);
       }
     }
   }
}
```

3.2 Computation of Betweenness Centrality

Once shortest path matrix is computed and stored for a given graph, it is then used to compute the betweenness centrality for each node. For a node i, the shortest path between each pairs of nodes $j \neq i$ and $k \neq i$ in the graph are analyzed and if node i lies on the shortest path from j to k, then number of such paths are calculated and divided by the total number of shortest paths between j and k. This is to be done for all nodes i in the graph. Pseudo code for this algorithm is shown below where *bet[i]* represents the betweenness of node i and *path_count(j, i)* represents the number of shortest paths from j to i.

```
Pseudo code for computation of Betweenness Centrality
for all nodes i
   for all pair of nodes (j,k) where shortest path from
       j to k  contains i
       bet[i] ← bet[i] + (path_count(j,i) ×
                      path_count(i,k))/path_count(j,k)];
}
```

4 Performance Improvement of the Algorithm

In order to improve the execution time of the algorithm given in section 3, we give a parallelized implementation of the algorithm that efficiently utilizes the resources of parallel architectures.

4.1 Implementation Using Multithreads

We implemented the algorithm using pthreads [8] in which the calculation of shortest path and betweenness centrality was divided into five threads. For the calculation of shortest path matrix, nodes of the input graph are equally divided among the threads created and each thread calculates the shortest path matrix of that many nodes only. After the shortest path matrix is calculated for each pair of nodes, we calculate the betweenness centrality. We again divide the nodes among threads to compute the betweenness centrality.

4.2 Implementation on Cluster Computer

Clustering is the use of multiple computers, multiple storage devices, and redundant interconnections, to form what appears to users as a single highly available system. It is used as a relatively low-cost form of parallel processing machine for scientific and other applications that lend themselves to parallel operations. MPI is used as a message-passing library specification.

First, the whole task was divided into several processes. The number of process was specified by the user while running the program. In our implementation we have used four processes. These processes were distributed among the cluster nodes. Then each process was divided into five threads. Thus the computation of betweenness centrality was first divided into different processes and then within each process, it was further divided into different threads. Each thread works on different node of the graph to compute the betweenness centrality.

5 Performance Analysis

We implemented our algorithm using C++. The system specification is HP xw4600 workstation with 3.0 GHz Core 2 Duo processor, and 3.48 GB RAM. We compared our algorithm given in section 3 with the algorithm proposed by Brandes[7].

5.1 Comparison of Our Algorithm with Brandes' Algorithm

We have implemented Brandes' algorithm [7] and our algorithm, as given in section 3, in *C++*. The graph in figure 4 shows the execution time of the algorithms for different number of nodes in a graph that was generated randomly.

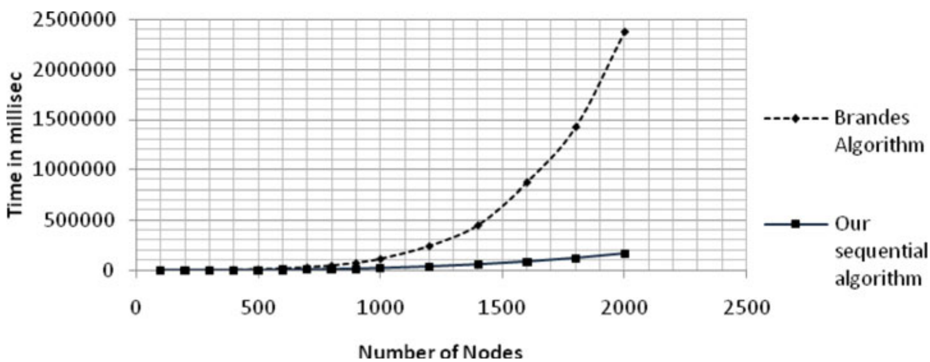


Fig. 4. Execution time of Brandes' and our algorithm

The worse case time complexity of Brandes' algorithm [7] and our algorithm is $O(n^3)$, when the number of edges is $O(n^2)$. However, Brandes' algorithm uses linear space. We have sacrificed space and thus achieved better execution time. Brandes' algorithm calculates the betweenness centrality by accumulating partial dependencies without saving the shortest distance for each pair of nodes. In our algorithm, first we store the shortest distance information in a matrix and then use it for the calculation of centrality. This increases the memory consumption of our algorithm but significantly reduces the time taken.

The time verses memory consumption graph comparing both the algorithms for 2000 nodes is shown in figure 5. It shows that the memory consumed by our algorithm increases with execution time as we are storing the shortest distance matrix. Then it becomes constant as the computation of shortest distance matrix is completed. As the cost of memory becomes cheaper day by day, our main concern remains the time performance. So in the tradeoff between time and memory, we chose time.

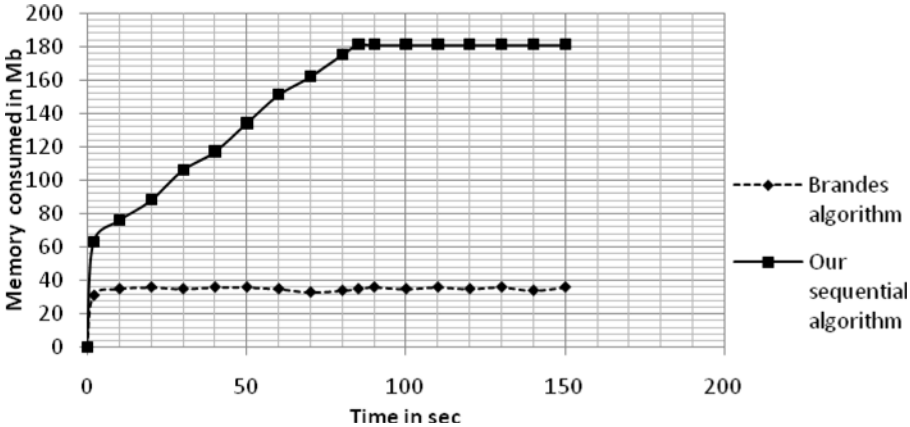


Fig. 5. Memory consumption of Brandes' and our algorithm

5.2 Performance Improvement Achieved Due to Parallelization

We compare the time taken by our sequential algorithm, multithreaded algorithm and parallel algorithm implemented on cluster computer. The graph in figure 6 shows the results of the comparison. The time taken by sequential algorithm is reduced by almost 50% in multithreaded algorithm and is further reduced by 20% in MPI cluster algorithm.

The pattern of the graph shows that the time performance gain will be more for large number of nodes as of the real online social network graphs. In figure 6, we have not compared the execution time of our implementations with that of Brandes' since it goes out of scope of the graph when number of nodes approach 2000. For instance, in figure 4, we find that when the number of nodes equal 2000, the time taken by Brandes' algorithm is nearly 2370 s which far exceeds the execution time of our algorithm which is nearly 166 s.

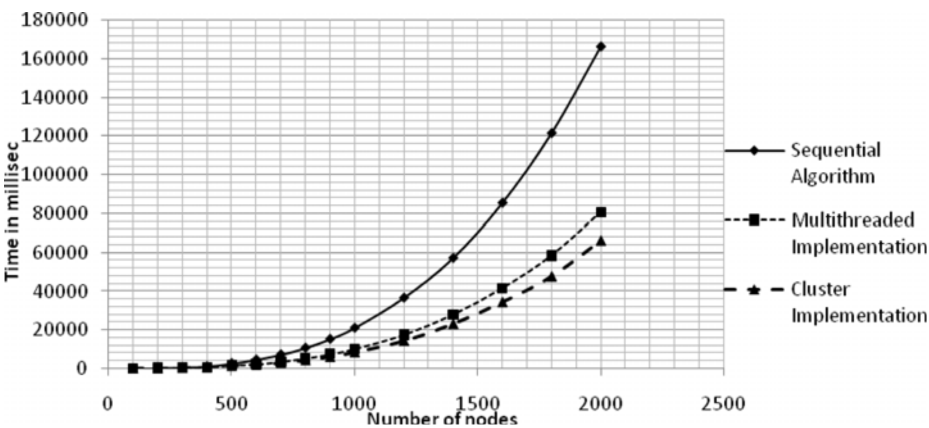


Fig. 6. Execution time of our sequential algorithm, multithreaded and cluster implementation

6 Related Work

We consider the recent developments made in the research on computation of betweenness centrality (BC). Brandes' algorithm [7] is an important contribution in this area. Till date, to the best of our knowledge, there is no algorithm better than that of Brandes' for the computation of BC. Moreover, there has not been much research attempts to develop better algorithms for the computation of BC since 2001. One reason might be that the approach in [7] suggests that there may be little scope for improving the time complexity, although no result on the lower bound to the problem was established. It is only very recently proved (in 2008) by Kintali [10] that any path-comparison based algorithm cannot compute BC in less than O(nm) time.

Our motivation for computation of betweenness centrality in a cost effective way comes from the fact that the size of memory is rapidly increasing and the cost of memory is decreasing day by day. Under these circumstances, we investigate how the computation of centrality measures can be done efficiently when space is not very significant. Although such an approach may not be attractive from the point of view of algorithm design, from a practical point of view we feel it may be useful and to what extent the results are improved is what we investigated in our experiments.

An approximation algorithm is given by Bader et al in [11]. The algorithm in [11] is based on an adaptive sampling technique. It has been tested on several real world data problems and the results are promising. Giesberger et al [12] suggests a framework for unbiased approximation of betweenness. Such an approximation has been empirically verified on many real world inputs.

Recent developments on high performance computing architectures have led to some work on designing parallel algorithms [13,14,15,16] for computing betweenness centrality. Our work differs from these as follows. Although we have obtained parallelization using multiple threads, we have not developed any parallel algorithm as in these works.

7 Conclusion and Future Work

The framework proposed in this paper provides a time efficient algorithm for the computation of different centrality measures. With our proposed algorithm, centrality of n node social network graph can be accurately computed where n is number of nodes that is scalable and user specified.

Our parallel algorithm uses resources like milticore processors and hence reduces the run-time significantly. We have also demonstrated successful use of cluster computer with MPI for the computation of betweenness centrality. The overall run time was drastically reduced as compared to the Brandes' algorithm.

We have performed our experiments by generating graphs of different size randomly.

As part of our future work we would like to test the performance of our algorithm on real online social network graphs. The algorithm can be further implemented on different architectures like CUDA to analyze the performance gain. Development of an efficient parallel algorithm and implementation on parallel architectures is another like of research that we would like to pursue.

As stated earlier, we compromised with the memory consumed for time performance gain. It would be interesting to design an algorithm for betweenness centrality that reduces the memory consumption without increasing the execution time.

References

1. HitWise Press Release (2006), `http://www.hitwise.com/press-center/hitwiseHS2004/social-networking-june-2006.php`
2. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications. Cambridge University Press, Cambridge (1994)
3. Freeman, L.C.: Centrality in social networks: Conceptual clarification. Social Networks 1, 215–239 (1979)
4. Li, L., Alderson, D., Doyle, J.C., Willinger, W.: Towards a Theory of Scale-Free Graphs: Definitions, Properties, and Implications. Internet Mathematics 2(4), 431–523 (2006)
5. Everett, M.G., Borgatti, S.P.: Krackhardt. D.: Ego-network betweenness. Paper presented at the 19th International Conference on Social Network Analysis (Sunbelt XIX), Charleston, South Carolina (1999)
6. Ellis, A.R.: Using SAS to Calculate Betweenness Centrality, Connections: International network For Social Network Analysis, vol. 29, pp. 26–32 (2009)
7. Brandes, U.: A Faster Algorithm for Betweenness Centrality. Journal of Mathematical Sociology 25(2), 163–177 (2001)
8. Nichols, B.: Pthreads Programming: a Posix Standard for Better Multiprocessing. Ignatius Press, San Francisco (1996)
9. Vetter, S., Aoyama, Y., Nakano, J.: Practical MPI Programming. IBM Redbooks publication (1999)
10. Kintali, S.: Betweenness Centrality: Algorithms and lower bounds, arXiv.org>cs>arXiv:0809.1906 (2008)
11. Bader, D.A., Kintali, S., Madduri, K., Mihail, M.: Approximating betweenness centrality. In: Bonato, A., Chung, F.R.K. (eds.) WAW 2007. LNCS, vol. 4863, pp. 124–137. Springer, Heidelberg (2007)
12. Griesberger, R., Sanders, P., Schultes, D.: Better approximation of betweenness centrality. In: Workshop of algorithm engineering and experiments (January 2008)
13. Bader, D.A., Madduri, K.: Parallel algorithms for evaluating centrality indices in real-world networks. In: Proc. 35th Int'l. Conf. on Parallel Processing (ICPP 2006), Columbus, Ohio, pp. 539–550. IEEE Computer Society, Los Alamitos (2006)
14. Madduri, K., Ediger, D., Jiang, K., Bader, D.A., Chavarria-Miranda, D.: A faster parallel algorithm and efficient multithreaded implementations for evaluating betweenness centrality on massive datasets. In: Proc. 3rd Workshop on Multithreaded Architectures and Applications (MTAAP 2009), Rome, Italy, May 2009, IEEE Computer Society, Los Alamitos (2009)
15. Tan, G., Tu, D., Sun, N.: A parallel algorithm for computing betweenness centrality. In: Intl. Conference on Parallel Processing, IEEE Xplore (2009)
16. Tu, D., Tan, G.: Characterizing betweenness centrality algorithm on multicore architectures. In: IEEE Intl. Symposium on Parallel and Distributed Processing with Applications (ISPA), Chengdu, China (August 2009)

Approaches to Preference Elicitation for Group Recommendation

Inma Garcia, Laura Sebastia, Sergio Pajares, and Eva Onaindia

Dpt. Computer Science, Universitat Politècnica de València
Camino de Vera s/n, E46022, Valencia, Spain
{ingarcia,lstarin,spajares,onaindia}@dsic.upv.es

Abstract. Recommendation can be defined as the problem of selecting, among a set of items, those ones that are likely of interest to the user. In case of a group of users, recommendations should satisfy, as far as possible, the preferences of all the group members. In order to elicit the group preferences, we present two different mechanisms: the first one consists in a voting procedure whereas the second is based on a negotiation procedure. In both cases, intelligent agents act on behalf of the group members.

The experimental results show the pros and cons of both approaches and highlight which of the two mechanisms returns the highest-valued recommendation for the whole group in each case. Moreover, we also study which approach is able to reflect more easily the different behaviour of each user, which is also an important aspect in group recommendation.

Keywords: Recommender Systems, Group Recommenders, Negotiation.

1 Introduction

Recommender systems (RS) are widely used in the internet for suggesting products, activities, etc. These systems usually give a recommendation for a single user considering his/her interests and tastes. However, many activities such as watching a movie or going to a restaurant, involve a group of users. In this case, the RS should take into account the preferences of all users in the group. The first task of this type of systems is to identify the individual preferences and then find a compromise that is accepted by all members in the group. This is the crucial point in a group RS because how individual preferences are managed to come up with group preferences will determine the success of the recommendation.

Over the last years, there have been several approaches to address this problem, most of them based on the combination of the individual preferences. The purpose is that of obtaining a recommendation that equally satisfies, as much as possible, all the group members according to their individual profiles. Although this approach is valid, it seems more realistic trying to capture the different attitudes of each member in the group with respect to the others. Specifically, in this paper, we present two different approaches to solve the recommendation problem: the first approach consists in a voting procedure, and the second one

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 547–561, 2011.

is based on a negotiation mechanism. Our proposal is to have a *user agent* that acts on behalf of a user of the group and communicates with the other agents to elicit the final recommendation. A user agent will first build the user preference model, which will contain the issues under negotiation.

In a voting (social choice) setting, all agents give input to a mechanism, and the outcome that the mechanism chooses based on these inputs, is a solution for all of the agents. In most settings, this outcome is enforced so that all agents have to abide to the solution prescribed by the mechanisms [12]. The classic goal has been to derive a social choice rule that ranks feasible social outcomes based on individuals' rankings of those outcomes. Our agent takes preference model of each each agent and it applies a protocol similar to the *plurality protocol* [12], which is a majority voting protocol where all the alternatives are compared simultaneously, and the one with the highest number of votes wins. Specifically, we apply an iterative plurality protocol in order to obtain a ranked list of outcomes, instead of obtaining a unique winner.

Negotiation in agent based systems can be defined as "*the process by which a group of agents come to a mutually acceptable agreement on some matter*" [5]. Negotiation is a natural and intuitive way of communication between agents acting as human beings [8]. As the behaviour of agents can be different from each other, negotiation allows a better catching of the group dynamics. Unlike other RS that use negotiation, and in which agents adopt the role of seller-buyer [7], in our system agents representing users adopt the same role because they have the same objective of finding a group recommendation. The negotiation process in our RS is based on the model of *alternating offers* [6], where an agent makes a proposal and each of the other agents may either accept the offer, reject it or opt out of the negotiation[1]. If an offer is accepted by all of the agents, the negotiation ends with an agreement. If at least one agent opts out the negotiation, then the negotiation terminates without reaching an agreement. If the offer is rejected by at least one agent, the negotiation proceeds to the next agent, who makes a new proposal. In our negotiation protocol, an agent has the possibility of accepting or rejecting part of the received proposal, which facilitates reaching an agreement in less time.

The paper is organized as follows. Section 2 gives a state of the art of group recommender systems. Section 3 describes the multi-agent system, the roles distinguished in the system, the ontology and the steps to obtain a group recommendation. Section 4 is devoted to the voting process, section 5 gives the details of the negotiation process and section 6 explains how the final set of items is selected. Section 7 shows the experiments performed to test our system. We finish with some conclusions in section 8.

2 Background

The main issue in a group RS is building a group profile that considers the tastes and preferences of the whole group. In the literature, we can find two

[1] For the sake of clarity, we restrict the negotiation to two agents.

approaches to this task: centralized and distributed. *Intrigue* [1] and *PolyLens* [9] are examples of the centralized approach. *Intrigue* assists a group of users in the organization of a tour. The final group recommendation does not consider individual participants one by one but the system models the group as a set partitioned into a number of homogeneous subgroups using socio-demographic information, and their possibly conflicting individual preferences are separately represented. *PolyLens* elaborates individual recommendations produced for each member of the group and merges these lists into a single ranked list according to the individual recommendation score and a social value function. These RS use, more or less sophisticated aggregation methods to compute the group profile.

As far as we know, there are two systems using a distributed approach. The *Travel Decision Forum* [4] allows the users to define their preferences and behaviour. It uses animated characters (avatars) to help the members of a group to agree on the organization of a vacation. The system uses a character that represents a mediator, who directs the interaction between the users. The mediator builds a proposal with the aggregation of the preferences of all users and asks each member of the group in turn whether the group model can be accepted or not. Avatars criticize the proposal according to their defined preferences and behaviour. In this system, only the mediator is able to compose the proposal. However, in our negotiation procedure the agents are in charge of building this proposal without a mediator.

The other similar approach is used in [3] for planning a trip for a group of users. It applies a cooperative negotiation methodology to group recommendation problem solution. Each user is represented by a negotiation agent, all with the same behaviour, that obtains the user recommendation by means of the case-based recommender system Trip@dvice [10]. Once the single user recommendations are obtained, the negotiation process starts. The system uses two negotiation protocols: mediated negotiation and direct negotiation. In the direct negotiation protocol, each negotiation participant has an opportunity to place an offer or accept one of the previously placed offers by the other negotiation participants. However, and as far as we know, this system does not provide mechanism to accept part of the offer or to make a counteroffer.

3 Description of the Multi-agent System

The aim of our Multi-Agent System -MAS- (figure 1) is to offer a recommendation to a group of users. We distinguish four different roles in this MAS:

– **User role:** Each user in the group is represented by an agent playing this role which is in charge of:
  1. **Building and updating the user profile**. When a user first enters the system, the first step is to register and enter his personal details and general likes. The user may also configure some aspects of the agent behaviour during the recommendation process.
  2. **Computing the model of the corresponding user**. The agent uses the underlying recommender system (GRSK [11]) to elicit the user preference model according out of his/her profile.

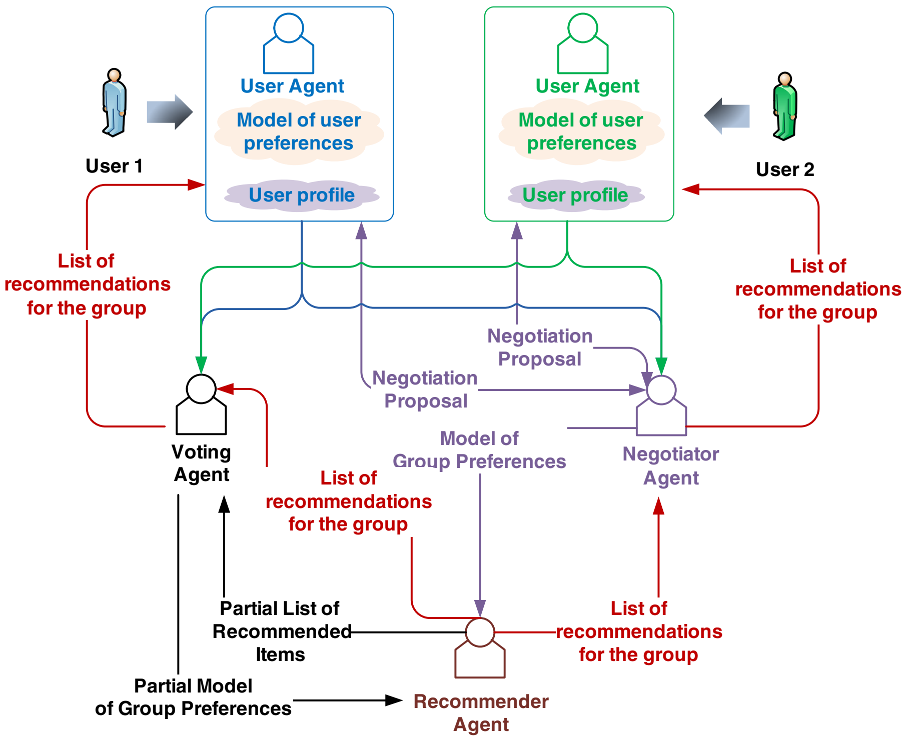


Fig. 1. Multi-agent system for group recommendation

3. **Participating in the voting/negotiation procedure.** Each agent knows both protocols and interacts with the other user agents and with the voting/negotiator agent to elicit group preference model.

- **Voting role:** The agent that plays this role (only one per group) takes as input the model of user preferences of each agent and applies an iterative plurality protocol to obtain the final list of recommendations for the group (with the help of the recommender agent).
- **Negotiator role:** The agent that plays this role (only one per group) is in charge of supervising the negotiation process. It starts the negotiation by sending a message to one of the user agents and it is informed when the negotiation ends. In case an agreement is reached, it requests the recommender agent to select the items of the recommendation.
- **Recommender role:** The agent playing this role (only one per group) takes as input the group preference model and retrieves the items that fulfill these preferences, which, in turn, will make up the final recommendation for the group.

During the group configuration, which consists in indicating which users will be part of the group, users will also decide on which of the two mechanisms (voting/negotiation) to use in order to elicit the group preference model.

3.1 Ontology

Our system uses an 'is-a' **ontology** to provide a common basis of understanding about the application domain. In this paper, we use a well-known movies domain, `MovieLens`[2]. This ontology defines 20 classes defining a list of genres. Although this ontology is quite simple, our system is prepared to use more complex ontologies.

Items are described by means of a list of tuples which represent all the classes that describe this item. In the remainder, a class describing an item is called a *feature*. A tuple is of the form (i, f, d^{if}), where i is the item, f is a feature defined in the ontology, and $d^{if} \in [0, 100]$ is the degree of interest of the item i under the feature f. Additionally, an item is associated a numeric value, denoted by RC^i, which represents how popular the item is among users.

For example, item *i=Cinema Paradiso* is described by the tuples {(i, comedy, 40), (i, drama, 40), (i, romance, 40)} and item *j=Chairman of the Board* is described by (j, comedy, 80), which indicates that the latter is more fun than *Cinema Paradiso*.

3.2 User Profile

User agents record a profile for each user u in the group, which is used to elicit the *user preference model*, a list of preferences as detailed in section 3.3. This profile records personal and demographic details about the user like the age, the gender, the family or the country; the general-likes model of the user, which is a description of the types of items the user u is interested in; and it also contains information about the historical interaction of the user with the RS, namely the set of items the user has been recommended and his degree of satisfaction with such recommendation.

3.3 Analysis of the User Profile

The first step towards the construction of the model of group preferences is to analyze the individual profiles with the aim of extracting the characteristics of the items that best suit each user. As a result, the user agent elicits a set of preferences, named the *user preference model* (P^u), which will be later used in the voting/negotiation procedure to determine the model of group preferences. Each preference is a tuple on the form (u, f, d^{uf}), where u denotes the user, f denotes a feature and d^{uf} indicates the estimated degree of interest of the user in feature f.

[2] www.grouplens.org

This elicitation process is performed by means of four basic recommendation techniques, namely demographic, content-based, collaborative and general likes-based techniques. These techniques receive the user profile and produce a list of individual preferences for each type of recommendation. After this phase, these lists are combined to obtain a single list, which describes the habitual tastes of the user.

3.4 Elicitation of the Model of Group Preferences

Once each user agent has elicited the model of preferences for his corresponding user, it is necessary to combine the preferences of all the group members to obtain the *model of group preferences*, denoted by P^G. Each element in P^G has the form (G, f, d^{Gf}), where G denotes the group, f denotes a feature and d^{Gf} indicates the estimated degree of interest of the group in feature f. As explained above, in our MAS we have developed two methods for deriving these group preferences: voting and negotiation. Next section details the first mechanism and section 5 explains the negotiation process.

4 Voting Process

The preferences predicted for the group are some function of all of the known preferences for every user in the group. Social Choice theorists, concerned with the properties of voting methods, have been investigating preference aggregation for decades. A very popular work is that of Arrow [2] which demonstrates the impossibility of combining individual preferences into a single expression of social preference in a way that satisfies several desirable properties. However, there are other investigations, specifically on collaborative filtering RS, that show that the only possible form for the prediction function is a weighted average of the users' ratings.

Voting procedures can be considered as *centralized* procedures, because all the agents give input to a mechanism which is in charge of combining all these inputs to obtain a solution for all the agents. In our case, the *voting agent* (one per group) supervises the voting procedure. Every user agent sends a message to the voting agent containing his user preference model; this message has the form `(inform :content` P^u`)`, where `inform` is a performative from FIPA-ACL[3]. The outcome of the voting agent will be the final list of recommendations. This list is obtained by means of a two-stage procedure, which first computes a weighted average of the most voted preferences among the users in the group, that is the preferences shared by the largest possible group of members and then retrieves the items that fulfill these preferences.

Specifically, the input of the voting procedure is the individual preference model P^u of every member in the group. The goal is calculating the final list of recommendations for the whole group. This is an iterative process, where we distinguish the following steps:

[3] FIPA-ACL Message Structure Specification.
http://www.fipa.org/specs/ fipa00061/SC00061G.html

Step 1: A *partial* group preference model P^G is elicited. At this stage, a plurality voting protocol [13] is used to select the most voted features. This way, a feature f is voted by a user if there is a preference for the feature f in his list P^u. We define $votes(f)$ as the number of users in the group whose P^u contains a preference over the feature f. At the first iteration, we *only*[4] select the features with the highest number of votes ($N = |G|$ votes). The value d^{Gf} associated to each feature is computed as the average of d^{uf} for all the users whose preferences include f.

$$P^G = \{(G, f, avg_{\forall u \in G}(d^{uf})) : votes(f) \geq N\}$$

Step 2: The items that satisfy this P^G are selected. The items to recommend will be selected by the recommender agent as explained in section 6.

If there are not enough items to cover the requested number of recommendations, at the next iteration, in Step 1, we select the features that received at least $N = |G| - 1$ votes, and so on. This way, we incrementally consider a more complete group preference model consisting of the features shared by the largest possible number of users in the group.

5 Negotiation Process

Let UA be the set of user agents involved in the recommendation process. After the first phase of the recommendation, all the agents in UA have computed the user models, P^{u_i}. Additionally, each group of users is associated a negotiator agent in charge of supervising the negotiation process. This section details the negotiation protocol and the internal behaviour of each user agent.

5.1 Negotiation Protocol

This MAS uses a direct synchronous protocol of negotiation. The strategic-negotiation model is based on the model of alternating offers. The issues under negotiation are the preferences that will be part of the group preference model. The communication between the agents in the system is established by exchanging messages. We use some performatives from FIPA-ACL. It is assumed that the agents can take actions in the negotiation only at certain times in the set $\mathcal{T} = \{0, 1, 2, \ldots\}$ that are determined in advance and known to the agents.

The negotiation process to calculate the preference model of the whole group starts when the negotiator agent sends a `(request :content "start negotiation")` message to one of the user agents in UA. In general, the negotiator agent participates in the negotiation process as a mediator: it receives the proposals of all the user agents and creates a common proposal that is then evaluated by the user agents. However, in this paper, for the sake of simplicity,

[4] For this reason, we say that this is a partial model of group preferences.

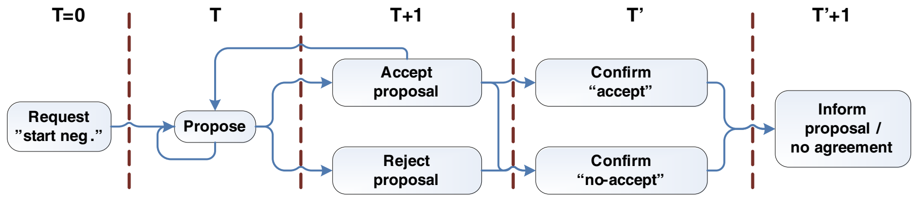


Fig. 2. Sequence of messages exchange

we assume that there are only two user agents participating in the negotiation process. For this reason, the negotiator agent only starts the negotiation process and it is informed when the negotiation ends, but it does not participate in the negotiation process. Therefore, we assume that only agents ua_1 and ua_2 belong to UA. If at time $t \in \mathcal{T}$, ua_1 sends a message to ua_2, the structure of this message is *(performative :content (response to the previous proposal, new proposal))*, where:

– the *performative* is one of the following FIPA-ACL performatives (with the usual meaning): `propose`, `accept-proposal`, `reject-proposal` and `confirm`.
– the *response to the previous proposal* (that is, the response to the proposal of agent ua_2 at time $t-1$) is a pair of lists of preferences of the form $\langle accepted_t, rejected_t \rangle$. $accepted_t$ is the list of preferences accepted by both agents, i.e. the list of preferences in $accepted_{t-1}$ plus those preferences proposed at $t-1$ by ua_2 and accepted by ua_1 at t. The list $rejected_t$ are those preferences proposed at $t-1$ and rejected by ua_1 at t.
– the *new proposal* is the list of preferences that compose the counteroffer of agent ua_1 to the previous proposal of ua_2 (denoted by $proposed_t$).

Figure 2 graphically shows the sequence of messages user agents exchange between each other. First, the negotiator agent selects one of the agents to start up the negotiation process, which receives a `request` message ($t = 0$). Then, ua_1 and ua_2 exchange as many `propose` messages as necessary ($t = 1, 2, \ldots$) until one of them (let's say ua_1) is satisfied with the *new proposal* or with part of it. In this situation, ua_1 sends an `accept-proposal` message to ua_2, with the part of the proposal that is satisfactory (in $accepted_{t+1}$). ua_2 may either: confirm the agreement (`confirm :content "accept"`), continue with the negotiation if the received proposal is not satisfactory (`propose`) or send a `confirm :content "no-accept"` message if the received proposal is not satisfactory and no other offers can be built. Finally, after the exchange of `propose` messages, ua_1 might want to finish the negotiation without agreement because it has no more proposals and all the previous ones have been rejected; in this case, ua_1 sends a `reject-proposal` message, which is answered by a `confirm :content "no-accept"` message from ua_2. The user agent that receives the last `confirm` message, is in charge of sending an `inform` message to the negotiator agent, indicating the result of the negotiation, that is, the final proposal accepted by both agents or that no agreement has been reached.

5.2 Behaviour of an Agent

The objective of the user agents is to compose a group preference model as closer to its own model of user preferences as possible. The user may configure some aspects of the user agent behaviour in order to facilitate to reach an agreement. We distinguish two different agent behaviours according to the degree of collaboration:

1. **Self-interested:** the agent will only consider as negotiation issues the preferences included in his own model of user preferences.
2. **Collaborative:** the agent considers as negotiation issues the preferences proposed by other agents in the hope that these agents will have the same behaviour.

Each user u defines a minimum threshold T_{min} (a value between 0 and 1) to help its user agent decide when a proposal is satisfactory enough to reach an agreement for u (self-interested behaviour). In case no agreement is reached with this threshold, the user may indicate that he wants the agent to carry out a collaborative negotiation. Specifically, the user establishes the number of preferences of the other users that can be proposed in each offer.

On the other hand, the user agents perform a *progressive negotiation process* over the list of preferences that describe the user model. For this reason, the preferences in P^u are organized in levels (named *levels of negotiation*), according to the degree of interest (d^{uf}) in the user model. The user may also define the values that determine each level of negotiation. Therefore, the user agents propose the preferences from the highest d^{uf} level to the lowest and, as the negotiation progresses, a greater number of preferences are considered in the offers.

The internal behaviour of a user agent is shown in figure 3. Assuming that ua_1 has received the `request` message from the negotiator agent, it must compose the first `propose` message. Recall that the structure of this message is: `(propose :content` $\langle accepted_t, rejected_t \rangle, proposed_t$`)`, where $accepted_t$, $rejected_t$ and $proposed_t$ are lists of preferences. In the first proposal ($t = 1$), both $accepted_t$ and $rejected_t$ are empty, whereas $proposed_t$ are the preferences in the first non-empty level of P^{u_1}.

This message is then evaluated by ua_2 to decide whether an agreement is reachable or not. It is important to remark at this point that a user agent may accept only a part of the received proposal. This way, a preference included in $proposed_t$ is accepted by ua_2 if it belongs to the user model (P^{u_2}) in a level higher or closer to his actual level of negotiation. Otherwise, p is rejected by this agent. The set of preferences proposed by ua_1 (that is, in $proposed_t$) and accepted by ua_2 is denoted by PA_t. Then, the new list of accepted preferences of ua_2 is composed by the preferences in $accepted_t$ (which were accepted in previous iterations of the negotiations) plus the preferences in PA_t (which have been accepted in the current iteration). At this moment, a scoring function is used to evaluate this list of accepted preferences. The scoring function $F^{ua_2}(t)$ of agent ua_2 over the list of accepted preferences at period t is defined as:

$$F^{ua_2}(t) = \frac{\sum\limits_{f \in accepted_t \cup PA_t} d^{u_1,f}}{\sum\limits_{f \in P^{u_1}} d^{u_1,f}}$$

where $d^{u_1,f}$ is the degree of interest for feature f in the user model P^{u_1}. That is, this scoring function computes the degree of the accepted preferences by the agent ua_2. In this situation, ua_2 accepts this new list of preferences if $F^{ua_2}(t) \geq T_{min}$. Otherwise, ua_2 will make up a new `propose` message with a counteroffer. Specifically, the new proposal is built as follows:

- $accepted_{t+1} = accepted_t \cup PA_t$
- $rejected_{t+1} = proposed_t - PA_t$
- $proposed_{t+1}$ are the preferences in the next non-empty level of P^{u_2} that have not been previously accepted by the other agent.

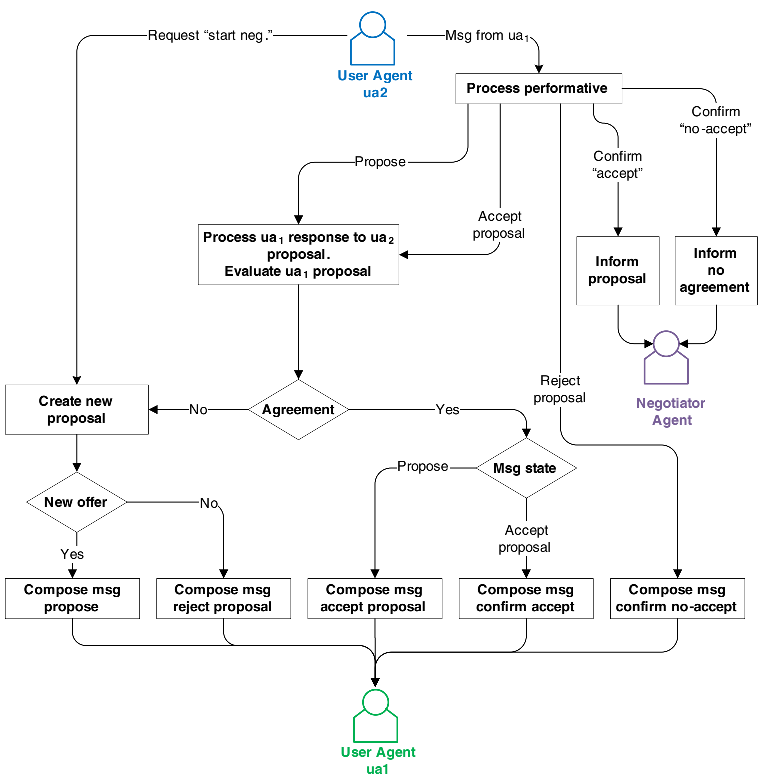


Fig. 3. Internal behaviour of a user agent

This self-interested behaviour may cause that an agreement cannot be reached if the users have heterogeneous tastes. For this reason, the user is allowed to configure the user agent behaviour to be more collaborative, as explained in section 3.2. In this case, the user agent builds proposals considering preferences of the other agent that were rejected before. $accepted_{t+1}$ and $rejected_{t+1}$ are computed as above; $proposed_{t+1}$ is composed of some preferences previously proposed by the other agent and that were rejected. The number of selected preferences is determined by the user in his negotiation profile. The agent selects those with the higher d^{uf} because it facilitates the agreement.

6 Recommendation of Items

As explained above, the recommender agent is in charge of retrieving the items that fulfill the preferences in the model of group preferences. In case the MAS is using the voting procedure to elicit this model, the recommender agent receives a `request` message from the voting agent every time this agent composes a partial model of group preferences. When using the negotiation procedure, the recommender agent receives a `request` message from the negotiator agent, once an agreement has been reached. Then, in both cases, the recommender agent receives a model of group preferences P^G as input.

An item described under a feature f is selected if there is a tuple (G, f, d^{Gf}) that belongs to P^G. The interest-degree of the group G in the item i, d^{Gi}, is calculated as follows:

$$d^{Gi} = percentile(RC^i) + avg_{\forall f}(d^{if} + d^{Gf})$$

where $percentile(RC^i)$ refers to the percentile rank of the positively rated counter of i (RC^i) with respect to the whole set of items positively rated by the users when i has been recommended. The second part of the formula considers the average interest-degree of every feature that describes the item i both in the ontology (d^{if}) and in the group preference model (d^{Gf}). The retrieved items make up the final recommendation for the group.

7 Experimental Results

In this section, we detail the experiments we carried out to analyze the voting and negotiation mechanisms. We have used the MovieLens data set with 943 users, 1682 movies and 100000 ratings.

The comparison of the performance of both mechanisms that is presented here is not aimed at revealing which mechanism outperforms the other, but at showing two different approaches for solving the problem. The voting technique is a single-agent centered approach which can be seen as an improvement of traditional techniques like the intersection or aggregation. On the other hand, the negotiation technique is a technique that involves as many agents as users in the group and is introduced as an attempt to simulate a human-like behaviour. We

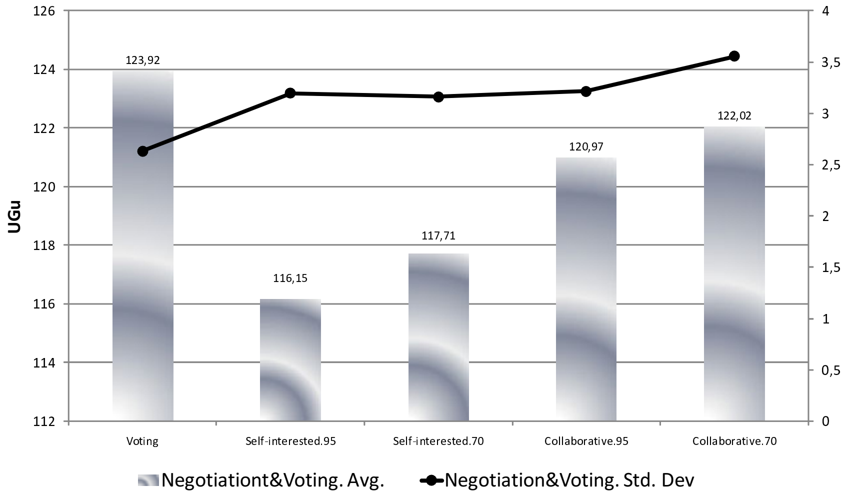


Fig. 4. Comparison of utility for each group (average and standard deviation) in voting and negotiation procedures

think that the negotiation technique presented in this paper is a good starting point to introduce other elements in group recommendation such as the level of trust of users in the other members of the group, or the incorporation of argumentative dialogues in the negotiation process.

Unlike individual recommendations, when dealing with groups, the most important issue is to obtain recommendations as satisfactory as possible for all the group members. Let RI^u be the recommendation for a single user u, such that each element in RI^u has the form (u, i, d^{ui}), where i is the recommended item and d^{ui} is the estimated interest degree of the user user u in the item i. RI^u contains a tuple (u, i, d^{ui}) for each item that can be recommended to user u, that is there is not a restriction on the number of items that are recommended to the individual user. If an item is not in RI^u, then it means the item has not been recommended to the user and thereby d^{ui} is set equal to 0. Given a recommendation RI^G of N items for a group G such that $u \in G$, the **utility** of such a recommendation for the user u is calculated as follows:

$$U_u^G = \frac{\sum_{\forall i \in RI^G} d^{ui}}{N}$$

This value gives a measure of the **precision** of the group recommendation for each user in the group because it takes into account how many items of the group recommendation would be recommended to the individual user. Moreover, U_u^G will be a higher value when the items in both the user and group recommendations are of more interest (higher d^{ui}) to the individual user. In order to analyze the quality of the recommendations, we consider the average and the standard deviation on the utility over all the group members. The average of the U_u^G

value gives a measure of the overall quality of the group recommendation. On the other hand, the standard deviation of U_u^G gives a measure of the satisfaction of each group member: if it is a low value, it means that U_u^G is quite similar for all the group members and no user is favored or prejudiced with respect to the others.

We have built 50 groups of two users by taking 100 random users from the MovieLens data set. First, the individual recommendation for each user has been elicited and then, the group recommendation for each group has been computed (N set to 20 items). For each group, we have calculated the utility of the group recommendation for each member (U_u^G). In the negotiation procedure, if no agreement is reached, the utility is set to 0.

First, we have executed the voting procedure and then the negotiation mechanism with four different configurations: (1) both user agents with a self-interested behaviour and with T_{min} set to 0.95, (2) both user agents with a self-interested behaviour and with T_{min} set to 0.70, (3) both user agents with a collaborative behaviour and with T_{min} set to 0.95 and (4) both user agents with a collaborative behaviour and with T_{min} set to 0.70. Figure 4 shows the results obtained in these experiments. Bars (referred to the scale on the left) represent the utility in average of the recommendations for each mechanism and configuration. The greatest utility value that can be obtained is 140. The line (referred to the scale on the right) determines the dispersion level in average for each mechanism.

If we compare the voting and the negotiation procedure, we can affirm that, as expected, the voting procedure gives better results than the negotiation because, as it is a centralized process, the preferences of all users are equally included in the group preference model. This way, the utility is higher and the standard deviation is lower.

Now we compare the results obtained in the different configurations of the negotiation mechanism. In case both agents exhibit a self-interested behaviour, the utility is greater when T_{min} is lower, because it is easier to reach an agreement (recall that no agreement means $U_u^G = 0$). In fact, when $T_{min} = 0.95$ an agreement is reached in the 94% of cases, whereas with $T_{min} = 0.70$ this value is increases up to 96%. With respect to the standard deviation, similar values are obtained in both cases. On the other hand, when both agents exhibit a collaborative behaviour, the best utility is obtained again with a lower T_{min}, whereas the standard deviation is lower with a higher T_{min}, which indicates that the users are more equally satisfied because the preferences included in the group preference model are closer to the highest user preferences.

As expected, we can also observe that it is easier to reach an agreement when both agents have a collaborative behaviour, because U_u^G is always equal or better compared to the self-interested behaviour with the same T_{min}. However, it is interesting to remark that the standard deviation is always better with the self-interested behaviour, which seems obvious, as the more collaborative an agent is, the more preferences not appearing in his user preference model will be included in the group preference model.

We also tested the negotiation mechanism with two user agents with different behaviours, but in most cases the result was that an agreement was easily reached. In the remaining cases, the agreement was not reached because the self-interested agent finished the negotiation immediately. The reason behind is that the MovieLens data set has a simple ontology (a list of genres) which make it difficult to have rich enough user profiles to execute complex negotiations.

As a conclusion, we can affirm that the voting procedure shows a better performance than the negotiation procedure from the point of view of the utility and standard deviation. However, negotiation is a more flexible approach which gives to the user the opportunity to configure a user agent closer to his real behaviour.

8 Conclusions

Recommendation can be defined as the problem of selecting, among a set of items, those ones that are likely of interest to the user. However, many activities involve a group of users. In this case, the RS must obtain the preferences that satisfy all the group members. To solve this problem, we propose a MAS where a user agent acts on behalf of one member of the group. This way, each user can define a different behaviour for the agent, which results in a more realistic social model.

We have implemented two protocols for the final recommendation: a voting protocol and a negotiation protocol. For the negotiation, we have defined three agent behaviours, according to the degree of collaboration, which determine the reasoning process to decide whether to accept or reject a proposal and to compose a new offer.

The experimental results show that the voting procedure exhibits a better performance than the negotiation procedure. However, negotiation is a more flexible approach which provides the user the opportunity to configure the user agent closer to his real behaviour.

We are also working on a new negotiation protocol based on the concept of "conditional negotiation". The general idea is to allow agents to exchange proposals in which an agent accepts some preferences provided that the other agent accepts some others. Additionally, we are working on incorporating techniques based on trust during the negotiation process, so that proposals are accepted or rejected depending on the proposer. This way, related users are favored to the detriment of others.

Acknowledgements. Partial support provided by Consolider Ingenio 2010 CSD2007-00022, Spanish Government Project MICINN TIN2008-6701-C03-01 and Valencian Government Project Prometeo 2008/051. FPU grant reference AP2009-1896 awarded to Sergio Pajares-Ferrando.

References

1. Ardissono, L., Goy, A., Petrone, G., Segnan, M., Torasso, P.: Intrigue: personalized recommendation of tourist attractions for desktop and handset devices. Applied AI, Special Issue on Artificial Intelligence for Cultural Heritage and Digital Libraries 17(8), 687–714 (2003)
2. Arrow, K.J.: Social Choice and Individual Values. Yale University Press, New Haven (1963)
3. Bekkerman, P., Kraus, S., Ricci, F.: Applying cooperative negotiation methodology to group recommendation problem. In: ECAI Workshop on Recommender Systems (2006)
4. Jameson, A., Baldes, S., Kleinbauer, T.: Enhancing mutual awareness in group recommender systems. In: Mobasher, B., Anand, E.S.S. (eds.) Proceedings of the International Joint Conference on Artificial Intelligence. Workshop on Intelligent Techniques for Web Personalization. AAAI, Acapulco (2003)
5. Jennings, N.R., Faratin, P., Lomuscio, A.R., Parsons, S., Wooldridge, M.J., Sierra, C.: Automated negotiation: Prospects, methods and challenges. Group Decision and Negotiation 10(2), 199–215 (2004)
6. Kraus, S.: Automated negotiation and decision making in multiagent environments. In: Luck, M., Mařík, V., Štěpánková, O., Trappl, R. (eds.) ACAI 2001 and EASSS 2001. LNCS (LNAI), vol. 2086, p. 150. Springer, Heidelberg (2001)
7. Lee, W.-P.: Towards agent-based decision making in the electronic marketplace: interactive recommendation and automated negotiation. Expert Systems with Applications 28(1) (2005)
8. Lenar, M., Sobecki, J.: Using recommendation to improve negotiations in agent-based systems. Journal of Universal Computer Science 13(2), 267–286 (2007)
9. O'Connor, M., Cosley, D., Konstan, J.A., Riedl, J.: Polylens: a recommender system for groups of users. In: European Conference on Computer Supported Cooperative Work, ECSCW 2001 (2001)
10. Ricci, F., Cavada, D., Nguyen, Q.N.: Integrating travel planning and on-tour support in a case-based recommender system. In: Proceedings of the Workshop on Mobile Tourism Systems (in conjunction with Mobile HCI 2002) (2002)
11. Sebastia, L., Garcia, I., Onaindia, E., Guzman, C.: e-Tourism: a tourist recommendation and planning application. International Journal on Artificial Intelligence Tools (WSPC-IJAIT) 18(5), 717–738 (2009)
12. Weiss, G.: Multiagent systems: a modern approach to distributed artificial intelligence. MIT Press, Cambridge (1999)
13. Wooldridge, M.: An introduction to multiagent systems. John Wiley & Sons, Chichester (2009)

Quantifying Downloading Performance of Locality-Aware BitTorrent Protocols

Lidong Yu, Ming Chen, and Changyou Xing

Department of Computer, PLA University of Science and Technology, Nanjing 210007, China
{yu.lidong,mingchennj,changyouxing}@163.com

Abstract. A plethora of locality-aware BitTorrent (BT) protocols have been proposed to mitigate cross-ISPs traffic and maintain comparable downloading performance as native BitTorrent. However, very limited effort has been paid to theoretically analyze locality-aware BitTorrent protocols, which may potentially make them limited significance for Internet traffic controlment. In this paper, aiming at providing a fundamental understanding of locality-aware BitTorrent protocols, a general analytical framework was proposed to analyze downloading performance of a large family of locality-aware BitTorrent protocols. Theoretical analysis and network measurement were carried out to quantify the metrics and properties of the analytical framework.

Keywords: Locality-aware BitTorrent protocol, downloading performance, geographical distribution of peers, analytical framework.

1 Introduction

Nowadays, BitTorrent (BT) file-sharing [1] has become one of the most successful peer-to-peer (P2P) applications over the Internet. However, downloading rate-based tit-for-tat (TFT) mechanism in BT results in a huge amount of intra- and inter-Internet Service Provider (ISP) traffic [2]. To address these issues, many locality-aware approaches [3-9] have been proposed.

Locality-aware is defined as the collection and usage of underlying Internet topology to guide BT peers to choose neighbors within particular neighborhood. It requires that BT peers know certain underlying information about all the other peers in the swarm of the sharing file. The underlying information of a host could be country/city locations, longitude and latitude, ISP locations (i.e., the identification of the ISP/autonomous system (AS), through which a host can connect to the Internet) and hops (e.g., IP hops, AS hops or PoP hops) etc. At present, it has been reported that locality-aware BitTorrent protocols [3-6, 9] can substantially reduce the costs of network provider, while, not dramatically degrading the performance of the native BT. In contrast to the aforementioned work, Piatek *et al.* [10] and Lehrieder *et al.* [11] demonstrated that a win-no lose situation between ISPs and BT users is practically difficult to achieve due to skewed peer distributions [8, 12] and heterogeneous upload capacity [13]. These two contrary conclusions are often confused. Since downloading performance of BT could be affected by many factors, e.g., heterogeneous upload capacities of peers, and geographical distribution of peers etc, researches favoring

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 562–576, 2011.

different factors may induce different conclusions. To the best of our knowledge, most of these locality-aware BitTorrent protocols ignore the geographical distribution of peers and have been reputably evaluated in controlled experimental environments. For example, P4P architectures [5] need ISPs and P2P applications to widely adopt them. It's a matter of fact that there is still a long way ahead of them. While Ono [6] always finds very few nearby neighbors in live swarms. A similar result was found in [10] that Ono can reduce inter-domain traffic by less than 1% when connecting to live swarms through a large residential ISP. The above examples demonstrate that an analytical framework analyzing and evaluating locality-aware BitTorrent protocols is crucial for the BT system performance.

Almost all these works [3-13] primarily focus on network measurement and simulation; much less has been done in developing a fundamental understanding of locality-aware BitTorrent protocols. Fan *et al.* [14] and Chow *et al.* [15] provided simple and complete steady state models of a heterogeneous BT system. They classified peers into several classes, where each class is characterized by its upload capacity. Qiu *et al.* [16] presented a simple fluid model to study the performance of BitTorrent. However, neither of them considered the neighbor-selecting diversities among peers, so their models are not applicable for the analysis of locality-aware BitTorrent protocol.

The aim of this paper is to provide a general analytical framework to analyze and evaluate downloading performance of locality-aware BitTorrent protocol. The main contributions are:

- **Downloading Rate Model of Locality-aware BT Client.** We propose a model of downloading rate of locality-aware BT. The model collectively considers the diversities of neighbor-selecting strategies (locality-aware BitTorrent and native BitTorrent), geographical distribution of peers (skewed peer distribution), and heterogeneous upload capacity of peers etc.
- **Simplex is Infeasible.** Simplex locality–aware neighbor-selecting strategy is defined as clients only adopting one locality–aware neighbor-selecting strategy, e.g., choosing all their neighbors from local ISP, AS, city, country, and regions of limited hops or distance etc. According to the Zipf-like geographical distribution of peers and heterogeneous upload capacity, geographical distributions of peers with heterogeneous upload capacities show great ununiformity. Even if the simplex locality–aware neighbor-selecting strategy is deployed on a large scale (e.g., all the clients in BT-like systems adopt it), one locality–aware BT client may potentially get poor downloading performance if too few neighbors upload data to it in its local neighborhoods.
- **Discriminative Effects on Downloading Rate.** Compared to leechers, the number of seeds and free-riders in the BT system will remarkably affect the downloading performance of clients. Increasing number of leechers in the BT system may not remarkably improve the performance of clients, since there are competitions among leechers.

The rest of the paper is organized as follows: Section 2 discusses related works. Section 3 illustrates analytical framework of locality-aware BitTorrent protocol. Section 4 is devoted to model validation. Finally, section 5 is the conclusion of the paper with further discussion of research directions.

2 Related Work

2.1 Overview of Native BT and Locality-Aware BT

To distinguish BitTorrent from locality-aware BitTorrent, BitTorrent is called native BT; locality-aware BitTorrent is called LocalBT in later sections. In native BT systems, nodes are organized in a swarm by sharing the same file. Nodes that have a complete copy of the file are termed *seeds*, otherwise they are termed *leechers*. If one leecher that does not contribute its uploading capacity is termed a *freerider*. Seeds, leechers, and freeriders are called as peers. Since each peer i, say P_i, is allowed to upload data (unchoke) to a fixed number of peers at a given time, it adopts TFT mechanism to *regular unchoke* other peers, i.e., those peers that currently offer the top highest x_r (default value is 4) uploading rate to it are regular unchoked. Additionally, x_o (default value is 1) peers are randomly selected for *optimistic unchoking*, which allows a peer to discover new mutually beneficial data exchange connections and bootstraps new comers.

However, downloading rate-based TFT mechanism in BT results in a huge amount of intra- and inter-ISP traffic. Many LocalBT mechanisms have been proposed. Karagiannis *et al.* [2] studied the potential benefits that locality-based solutions had for the content provider and the clients. Bindal *et al.* [3] proposed a mechanism that a peer chose majority of its neighbors from the same ISP by exploring biased neighbor selecting. Results show that a large fraction of the inter-AS traffic can be saved and that the median of the download times are decreased. Aggarwal *et al.* [4] proposed a solution where the ISP offers an "oracle" to the P2P users. When a P2P user fed the oracle with a list of possible P2P neighbors, the oracle ranked them according to certain criteria. Xie *et al.* [5] suggested an architecture called P4P, which provided multiple interfaces to networks for communication with applications. P4P improved P2P completion time by approximately 20% on average. Choffnes *et al.* [6] presented a scalable approach to biased peer selecting called Ono, which was based on CDN redirections. Agarwal *et al.* [7] designed Htrae, whose novel feature was its synthesis of geolocation information with a network coordinate system. Ren *et al.* [9] proposed a topology-aware BitTorrent client exploit $d/(u{*}l)$ for peer selecting.

However, most of these approaches have been reputably evaluated in controlled experimental environments, which might artificially improve the performance of locality-aware BitTorrent protocol. In contrast to these works, Piatek *et al.* [10] demonstrated that a win-no lose situation between ISPs and BT users was difficult to achieve in practice due to limited impact, reduced performance and robustness, and conflicting interests. Lehrieder *et al.* [11] presented similar results as [10] due to skewed peer distributions [8, 12] and heterogeneous access bandwidth [13] etc.

2.2 Skewed Geographical Distribution of BT Peers in Country and AS Level

Wang *et al.* [8] examined the content and the peer diversities in BT and demonstrated a Mandelbrot-Zipf distribution of peers in AS level. Further in [12], torrents were classified into two categories according to the language popularity of contents, i.e., popular language, and unpopular language. Language popularity is defined as the number of countries that take the language as their native language. If the popularity exceeds a threshold, the language is defined as a popular language, otherwise an unpopular one. For example, according to the definition of language popularity, Chinese is an unpopular language; English is a popular one. Yu *et al.* [12] presented that both in country level and AS level, peers in unpopular language-file swarms are fitted in Zipf-like distribution; while peers in popular language-file swarms fitted in Mandelbrot-Zipf distribution. Moreover, peers are more apt to cluster into few countries/ASes when language popularity declines. However, most of the locality-aware mechanisms [2-7] have failed to consider the geographical distributions of peers in real BT systems or constructed unreal experimental scenarios.

2.3 Model of Native BitTorrent

Piatek *et al.* [13] examined 301,595 unique BT IP addresses over a 48 hour period from 3,591 distinct ASes across 160 countries and observed that the upload capacity of 65% of peers was between 30-100KB/s, 25% of peers was between 100-800KB/s, and less than 10% of peers was over 800 KB/s.

A number of research studies (e.g., [14-16]) have focused on the fairness, robustness, and performance characteristics of BT. However, none of them considered the neighbor-selecting diversities among peers, and were not applicable to the analysis of locality-aware BitTorrent protocol. In the paper, an analytical framework was proposed to analyze locality-aware BT.

3 Analytical Framework of LocalBT

In this section, we will propose an analytical framework of LocalBT, which collectively considers the diversities of neighbor-selecting strategies, geographical distribution of peers, and heterogeneous upload capacity of peers etc.

3.1 Description of the Analytical Framework

3.1.1 Heterogeneous Upload Capacity

In BT systems, a set of peers $\mathbb{N}(t)=\{P_1,P_2,\cdots P_i,\cdots,P_n\}$ ($\mathbb{N}$ for short) that are simultaneously interested in one file are organized in a swarm $\boldsymbol{S}$ at time t. A peer i, say P_i, has its uploading rate denoted as u_i. We assume these peers have the following relationship in their uploading rates:

$$u_1 > u_2 > \ldots > u_m > u_{m+1} = \ldots = u_n = 0, \text{ for } m<n,\ i=1,2,\ \ldots,n\ . \tag{1}$$

$\mathbb{N}^{freeRider}=\{P_{m+1},P_{m+2},\cdots P_n\}$ denotes the set of freeriders whose uploading rates to the swarm are 0.

3.1.2 Diversities of Neighbor Selecting Mechanisms

At present, there have been many neighbor selecting mechanisms in BT-like systems, e.g., ISPbia [3], city-bias, country-bias, TopBT [9], and native BT etc. To understand how LocalBT affects the downloading performance of LocalBT clients, we assume that all LocalBT peers adopt the same neighbor selecting mechanism. For example, they can select all their neighbors from the same ISP/AS, country, or limited AS/IP hops etc. The rest peers in the swarm all adopt native BT protocol. Consequently, peers in the swarm can be classified into two categories:

$$\begin{aligned}\mathbb{N} &= \mathbb{N}^{L} \bigcup \mathbb{N}^{\bar{L}} = (\mathbb{N}^{Lseed} \bigcup \mathbb{N}^{Lleecher}) \bigcup (\mathbb{N}^{\bar{L}seed} \bigcup \mathbb{N}^{\bar{L}leecher} \bigcup \mathbb{N}^{freeRider}) \\ &= (\mathbb{N}^{Lseed} \bigcup \mathbb{N}^{\bar{L}seed}) \bigcup (\mathbb{N}^{Lleecher} \bigcup \mathbb{N}^{\bar{L}leecher}) \bigcup \mathbb{N}^{freeRider} \\ &= \mathbb{N}^{seed} \bigcup \mathbb{N}^{leecher} \bigcup \mathbb{N}^{freeRider}\end{aligned} \quad . \tag{2}$$

Here, $\mathbb{N}^{L}$ and $\mathbb{N}^{\bar{L}}$ respectively denote the sets of peers exploiting LocalBT protocol and native BT protocol. It is also assumed that one peer can only adopt one neighbor selecting mechanism, i.e., $\mathbb{N}^{L} \bigcap \mathbb{N}^{\bar{L}} = \varnothing$. $\mathbb{N}^{Lseed}$ and $\mathbb{N}^{Lleecher}$ respectively denote the set of seeds and leechers exploiting LocalBT protocol. $\mathbb{N}^{\bar{L}seed}, \mathbb{N}^{\bar{L}leecher}$ and $\mathbb{N}^{freeRider}$ denote the set of seeds, leechers and freeriders exploiting native BT protocol respectively. Specifically freeriders are assumed to exploit native BT protocol, which is true in that they just expect to completely download data as soon as possible. Since each peer can only behave as one role in anytime within $[t, t+\varepsilon)$, we have $\mathbb{N}^{seed} \bigcap \mathbb{N}^{leecher} = \varnothing, \mathbb{N}^{seed} \bigcap \mathbb{N}^{freerider} = \varnothing, \mathbb{N}^{leecher} \bigcap \mathbb{N}^{freerider} = \varnothing$

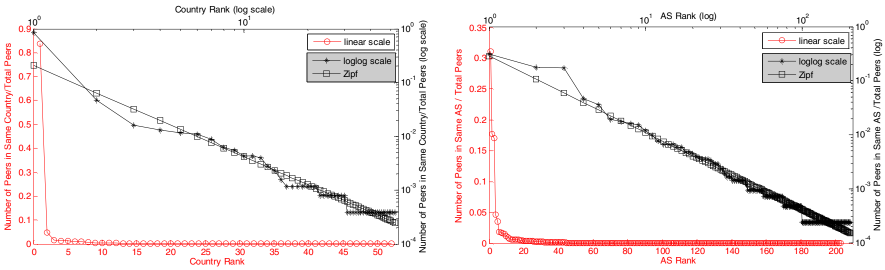
3.1.3 Geographical Distribution of Peers

Fig. 1(a) and Fig. 1(b) show that the geographical distributions of peers follow Zipf-like distribution both in country and AS level respectively [12] which means that majority of peers within any BT swarm are distributed in very few countries or ASes. This distribution is much relevant to the language of the shared file and the population of different ISPs/ASes. We say that a node is on *sparse mode* with respect to there are very few peers within its local neighborhood $\Re$ (the region may be a country-scale, AS-scale, or limited AS/IP hops etc). Contrarily the node is in *dense mode*. According to the skewed geographical distribution of peers, most BT peers are in dense mode.

Let $\mathbb{N}_{iR}$ be the set of peers in the same local region $\Re$ as peer *i*, e.g., within the same country or AS. Inversely $\mathbb{N}_{i\bar{R}}$ denotes the set of peers beyond the local region of peer *i*. Within local region $\Re$ of peer *i*, seeds $\mathbb{N}_{iR}^{seed}$ are composed of LocalBT seeds $\mathbb{N}_{iR}^{Lseed}$ and native BT seeds $\mathbb{N}_{iR}^{\bar{L}seed}$; leechers $\mathbb{N}_{iR}^{leecher}$ consist of LocalBT leechers $\mathbb{N}_{iR}^{Lleecher}$ and native BT leechers $\mathbb{N}_{iR}^{\bar{L}leecher}$. $\mathbb{N}_{iR}^{jLseed} = \{P_j \mid P_j \in \mathbb{N}_{iR}^{Lseed}\}, \mathbb{N}_{iR}^{j\bar{L}seed} = \{P_j \mid P_j \in \mathbb{N}_{iR}^{\bar{L}seed}\}$, $\mathbb{N}_{iR}^{jLleecher} = \{P_j \mid P_j \in \mathbb{N}_{iR}^{Lleecher}\}, \mathbb{N}_{iR}^{j\bar{L}leecher} = \{P_j \mid P_j \in \mathbb{N}_{iR}^{\bar{L}leecher}\}$, for $j=1,2,\cdots,n$. Since peer *j*, say P_j, can only behave as one role in anytime within $[t, t+\varepsilon)$, the potential values that $N_{iR}^{jLseed}, N_{iR}^{j\bar{L}seed}, N_{iR}^{jLleecher}$, and $N_{iR}^{j\bar{L}leecher}$ can take are 0 or 1 and $N_{iR}^{jLseed} + N_{iR}^{j\bar{L}seed} + N_{iR}^{jLleecher} + N_{iR}^{j\bar{L}leecher} = 1$.

3.2 Downloading Rate Model of LocaBT Clients

In this section, we propose a downloading rate model for LocalBT clients. In this model, our objective is to understand how the LocalBT protocol affects the downloading performance of LocalBT clients. Hence, we assume LocalBT clients just upload



(a) Peer distribution in country level

(b) Peer distribution in AS level

Fig. 1. Geographical distribution of peers in country level and AS level

and download data within their local region $\Re$. We also assume that all peers are fully connected and have demands from each other.

The downloading rate $d_i(t)$ of peer i maybe a sum of uploading rate from all leechers and seeds to it. So, the rate balance model is:

$$d_i(t)=\sum_{j=1}^{n} u_j^i N_{iR}^{j\bar{L}seed}+\sum_{j=1}^{n} u_j^i N_{iR}^{jLseed}+\sum_{j=1}^{n} u_j^i N_{iR}^{j\bar{L}leecher}+\sum_{j=1}^{n} u_j^i N_{iR}^{jLleecher} \tag{3}$$

Here, u_j^i is the uploading rate from peer j to peer i. $\forall P_i \in \mathbb{N}^{seed}, u_j^i=0, d_i(t)=0$. $\forall P_i \in \mathbb{N}^{freerider}, u_j^i=0$. Further $d_i(t)$ consists of three parts: seed unchoked rate, optimistic unchoked rate and regular unchoked rate from leechers, i.e., $d_i(t)=d_i^{seed}(t)+d_i^{reg}(t)+d_i^{opt}(t)$, which is similar to [15].

Seed Unchoked Rate

According to the results in [17], fairness could be achieved by a seed giving the same service time to each leecher or freerider. Videlicet, seeds upload uniformly to all leechers and freeriders. Within local region $\Re$, the probability that LocalBT leecher i is selected by native BT seeds $\mathbb{N}_{iR}^{\bar{L}seed}$ is $(x_s \cdot N_{iR}^{\bar{L}seed})/(N^{\bar{L}leecher}+N^{freeRider}+N_{iR}^{Lleecher})$. Here, x_s (default value is 5) is the number of simultaneous unchoked leechers or freeriders performed by a seed. Since $\mathbb{N}_{iR}^{Lseed}$ just uploads data to local peers, the probability that leecher i is selected by LocalBT seeds $\mathbb{N}_{iR}^{Lseed}$ is $(x_s \cdot N_{iR}^{Lseed})/(N_{iR}^{\bar{L}leecher}+N_{iR}^{freeRider}+N_{iR}^{Lleecher})$. So the seed unchoked rate of LocalBT leecher i is:

$$d_i^{seed}(t)=\sum\nolimits_{j=1}^{n} u_j\left(\frac{N_{iR}^{j\bar{L}seed}}{N^{\bar{L}leecher}+N^{freeRider}+N_{iR}^{Lleecher}}+\frac{N_{iR}^{jLseed}}{N_{iR}^{\bar{L}leecher}+N_{iR}^{freeRider}+N_{iR}^{Lleecher}}\right) \tag{4}$$

Optimistic Unchoked Rate

To evaluate the service capacity of other peers and bootstrap new comers, It assumes that the probability of every leecher or freerider optimistically unchoked by other leechers follows the uniform distribution. Within local region $\Re$, total optimistic unchoke capacities from LocalBT leechers are $\sum_{j=1}^{n} u_j \cdot N_{iR}^{jLleecher} \cdot x_o / (x_r + x_o)$; total optimistic unchoke capacity from native BT leechers are $\sum_{j=1}^{n} u_j \cdot N_{iR}^{j\bar{L}leecher} \cdot x_o / (x_r + x_o)$. The potential customers of $\mathbb{N}_{iR}^{Lleecher}$ are $\mathbb{N}_{iR}^{\bar{L}leecher} \bigcup \mathbb{N}_{iR}^{Lleecher} \bigcup \mathbb{N}_{iR}^{freeRider}$. While, for $\mathbb{N}_{iR}^{\bar{L}leecher}$, their potential interested customers might be $\mathbb{N}^{\bar{L}leecher} \bigcup \mathbb{N}^{freeRider} \bigcup \mathbb{N}_{iR}^{Lleecher}$. According to the rate balance law, the optimistic unchoked rate of LocalBT leecher i is:

$$d_i^{\mathrm{opt}}(t) = \sum_{j=1}^{n} u_j \cdot \frac{x_o}{x_r + x_o} \cdot \left(\frac{N_{iR}^{jLleecher}}{N_{iR}^{\bar{L}leecher} + N_{iR}^{freeRider} + N_{iR}^{Lleecher}} + \frac{N_{iR}^{j\bar{L}leecher}}{N^{\bar{L}leecher} + N^{freeRider} + N_{iR}^{Lleecher}} \right) \tag{5}$$

Apparently, as items of denominator in equation (4) and (5), increasing the number of freeriders $N_{iR}^{freeRider}$ and $N^{freeRider}$ will decrease $d_i^{seed}(t)$ and $d_i^{\mathrm{opt}}(t)$.

Regular Unchoked Rate

According to native BitTorrent protocol [1], anytime within $[t, t+\varepsilon)$, the maximum number of leechers simultaneously unchoked by leecher i is x_r+x_o. x_r and x_o respectively denote the number of simultaneous regular and optimistic unchoked leechers performed by a leecher.

Let $\mathbb{N}_i^U = \{P_j \,|\, u_i^j > 0,\ i \neq j,\ j = 1, 2, \cdots, n\}$ be the set of peers unchoked by leecher i. $\mathbb{N}_j^U$ denotes the set of peers unchoked by leecher j. We have $0 < N_i^U = |\mathbb{N}_i^U| \le x_r + x_o, 0 < N_j^U = |\mathbb{N}_j^U| \le x_r + x_o$. According to the tit-for-tat mechanism in native BT, the regular unchoked rate for peer i must come from peers belonging to $\mathbb{N}_i^U$. Sine LocalBT leechers are assumed to upload and download data within local region, $\mathbb{N}_i^U \subset \mathbb{N}_{iR}$. $\mathbb{N}_j^D$ denotes the set of peers that are unchoking leecher j. So, the regular unchoked rate for LocalBT leecher i is:

$$\begin{aligned} & d_i^{reg}(t) = \sum_{P_j \in \mathbb{N}_i^u} \frac{1}{N_j^U} \cdot u_j \cdot I_j^i \\ & s.t. \quad I_j^i = \begin{cases} 1 & \text{if} \quad 1 \le rank_i(u_i^j, u_k^j) \le x_r,\ P_k \in \mathbb{N}_j^D \setminus \{i\} \\ 0 & \text{else} \end{cases} \end{aligned} \tag{6}$$

The function of $rank_i(u_i^j, u_k^j)$ is to compare the uploading rate of leecher i with other leechers $k, P_k \in \mathbb{N}_j^D \setminus \{i\}$. If u_i^j ranks the top x_r in $\mathbb{N}_j^D$, leecher j will regular unchoke leecher i, otherwise leecher j will choke leecher i.

Proposition 1. For $\forall P_i \in \mathbb{N}_j^D, \forall P_k \in \mathbb{N}_j^D \setminus \{i\}$, when $1 \le rank_i(u_i^j, u_k^j) \le x_r$, the upper bound of $d_i^{reg}(t)$ is $\sum_{P_j \in \mathbb{N}_i^u} \frac{1}{N_j^U} \cdot u_j$. When $rank_i(u_i^j, u_k^j) > x_r$, the lower bound of $d_i^{reg}(t)$ is 0.

Proof: If $1 \le rank_i(u_i^j, u_k^j) \le x_r$, we get $I_j^i = 1$. According to equation (6), the upper bound of $d_i^{reg}(t)$ is $\sum_{P_j \in \mathbb{N}_i^u} \frac{1}{N_j^U} \cdot u_j = \sum_{P_j \in \mathbb{N}_i^u} \frac{1}{x_r + x_o} \cdot u_j$. If $rank_i(u_i^j, u_k^j) > x_r$, we get $I_j^i = 0$ and the lower bound of $d_i^{reg}(t)$ is 0. In real BT-like systems, $\mathbb{N}_i^U$ and $\mathbb{N}_j^U$ are usually fixed numbers, i.e., x_r+x_o (default value is 5). According to equation (6), it can be concluded that if one peer want to get high regular unchoked rate it should increase its uploading rate to other peers. This mechanism encourages BT peers to contribute their uploading capacities to the swarm. The proof is therefore complete.

3.2.1 Downloading Rate of LocalBT Leechers

The total downloading rate of leecher i is the sum of equation (4), (5), (6):

$$\begin{aligned} d_i(t) = d_i^{reg}(t) + d_i^{seed}(t) + d_i^{opt}(t) = \sum_{P_j \in \mathbb{N}_i^u} \frac{1}{N_j^U} \cdot u_j \cdot I_j^i + \\ \sum\nolimits_{j=1}^{n} u_j \left(\frac{N_{iR}^{j\bar{L}seed}}{N^{\bar{L}leecher} + N^{freeRider} + N_{iR}^{Lleecher}} + \frac{N_{iR}^{jLseed}}{N_{iR}^{\bar{L}leecher} + N_{iR}^{freeRider} + N_{iR}^{Lleecher}} \right) + \\ \sum\nolimits_{j=1}^{n} u_j \cdot \frac{x_o}{x_r + x_o} \cdot \left(\frac{N_{iR}^{jLleecher}}{N_{iR}^{\bar{L}leecher} + N_{iR}^{freeRider} + N_{iR}^{Lleecher}} + \frac{N_{iR}^{j\bar{L}leecher}}{N^{\bar{L}leecher} + N^{freeRider} + N_{iR}^{Lleecher}} \right) \end{aligned} \tag{7}$$

Let $\alpha = N^{freeRider}$, $\gamma = N_{iR}^{freeRider} / N^{freeRider}$, $\beta = N_{iR}^{Lleecher} / N^{freeRider}$, $\theta = N_{iR}^{\bar{L}leecher} / N^{freeRider}$ $\lambda = N_{i\bar{R}}^{\bar{L}leecher} / N_{iR}^{\bar{L}leecher}$, $\tau = N_{iR}^{\bar{L}seed} / N^{freeRider}$, $C = x_o / (x_r + x_o)$, $\psi = N_{iR}^{Lseed} / N^{freeRider}$, $\varphi = N_{iR}^{\bar{L}leecher} / N_{iR}^{Lleecher}$, where $\alpha, \beta, \theta, \lambda, \psi \ge 0$, $0 \le \gamma \le 1, 0 < C < 1$, the equation (7) can be rewritten as

$$\begin{aligned} d_i(t) = \sum_{P_j \in \mathbb{N}_i^u} \frac{1}{N_j^U} \cdot u_j \cdot I_j^i + \sum\nolimits_{j=1}^{n} u_j \cdot \Big(\frac{N_{iR}^{j\bar{L}seed}}{\alpha \cdot \theta + \alpha \cdot \lambda + \alpha + \alpha \cdot \beta} + \frac{N_{iR}^{jLseed}}{\alpha \cdot \theta + \alpha \cdot \gamma + \alpha \cdot \beta} + \\ \frac{C \cdot N_{iR}^{jLleecher}}{\alpha \cdot \theta + \alpha \cdot \gamma + \alpha \cdot \beta} + \frac{C \cdot N_{iR}^{j\bar{L}leecher}}{\alpha \cdot \theta + \alpha \cdot \lambda + \alpha + \alpha \cdot \beta} \Big) \end{aligned} \tag{8}$$

Let $\tilde{u}_{\bar{L}seed} = \sum\nolimits_{j=1}^{n} u_j \cdot N_{iR}^{j\bar{L}seed} / N_{iR}^{\bar{L}seed}$, $\tilde{u}_{Lseed} = \sum\nolimits_{j=1}^{n} u_j \cdot N_{iR}^{jLseed} / N_{iR}^{Lseed}$, $\tilde{u}_{\bar{L}leecher} = \sum\nolimits_{j=1}^{n} u_j \cdot N_{iR}^{j\bar{L}leecher} / N_{iR}^{\bar{L}leecher}$, $\tilde{u}_{Lleecher} = \sum\nolimits_{j=1}^{n} u_j \cdot N_{iR}^{jLleecher} / N_{iR}^{Lleecher}$, the equation (8) could be rewritten as:

$$d_i(t) = \sum_{P_j \in \mathbb{N}_i^u} \frac{1}{N_j^U} \cdot u_j \cdot I_j^i + \frac{\tilde{u}_{\bar{L}seed} \cdot \tau + \tilde{u}_{\bar{L}leecher} \cdot \theta \cdot C}{\theta + \lambda + 1 + \beta} + \frac{\tilde{u}_{Lseed} \cdot \psi + \tilde{u}_{Lleecher} \cdot \beta \cdot C}{\theta + \gamma + \beta} \tag{9}$$

Since τ and ψ are only present in the numerator in (9), increasing the values of τ and ψ will improve $d_i(t)$. It implies that if there are more seeds in local region joining the swarm, LocalBT leechers can get faster downloading rate. Since γ is only present in the denominator of equation (9), increasing the values of γ will degrade $d_i(t)$. It means that increasing the number of local freeriders will decrease the downloading rate of LocalBT leechers.

Proposition 2. when $\frac{\tilde{u}_{\bar{L}seed}}{\tilde{u}_{Lseed}} \leq \frac{\theta+\lambda+1+\beta}{\theta+\gamma+\beta}$, compared to native BT seeds, LocalBT seeds will devote more service capacity to LocalBT leechers. Contrarily, native BT seeds will contribute more than LocalBT seeds to LocalBT leechers.

Proof: Since we have analyzed the regular unchoke rate in Proposition 1, we will focus on the latter two items in equation (9). Let $y = \frac{\tilde{u}_{\bar{L}seed} \cdot \tau + \tilde{u}_{\bar{L}leecher} \cdot \theta \cdot C}{\theta+\lambda+1+\beta} + \frac{\tilde{u}_{Lseed} \cdot \psi + \tilde{u}_{Lleecher} \cdot \beta \cdot C}{\theta+\gamma+\beta}$, $\frac{\partial y}{\partial \tau} - \frac{\partial y}{\partial \psi} = \frac{\tilde{u}_{\bar{L}seed}}{\theta+\lambda+1+\beta} - \frac{\tilde{u}_{Lseed}}{\theta+\gamma+\beta}$. Since $0 \leq \gamma \leq 1$, then $\theta+\lambda+1+\beta > \theta+\gamma+\beta$. If $\frac{\tilde{u}_{\bar{L}seed}}{\tilde{u}_{Lseed}} \leq \frac{\theta+\lambda+1+\beta}{\theta+\gamma+\beta}$, then $\frac{\partial y}{\partial \tau} - \frac{\partial y}{\partial \psi} \leq 0$, otherwise $\frac{\partial y}{\partial \tau} - \frac{\partial y}{\partial \psi} > 0$. That is to say, when $\frac{\tilde{u}_{\bar{L}seed}}{\tilde{u}_{Lseed}} \leq \frac{\theta+\lambda+1+\beta}{\theta+\gamma+\beta}$, compared to traditional BT seeds, LocalBT seeds will devote more service capacity to LocalBT leechers. Contrarily, traditional BT seeds will contribute more than LocalBT seeds to LocalBT leechers. The proof is therefore complete.

Propositions 3. Increasing the number of native BT leechers beyond the region $\Re$ will degrade the downloading rate of LocalBT leechers. Increasing the number of native BT leechers within the region $\Re$ may potentially degrade the downloading rate of LocalBT leechers.

Proof: Since λ is present in the denominator of equation (9), increasing the values of λ will degrade $d_i(t)$. The possible reason is that increasing the λ results in increasing the competitions among leechers.

According to the skewed geographical distribution of peers and real deployment of LocalBT protocol in BT systems, for most countries or ASes, $\alpha, \theta \geq 1,\ 0 \leq \beta, \gamma, \lambda, \psi, \tau \leq 1$, we can easily get $\frac{\partial y}{\partial \theta} < 0$, which means that increasing the number of traditional BT lechers within the region $\Re$ may potentially degrade the downloading rate of LocalBT leechers. Previous research [18] holds that increasing the number of peers will improve the service capacity of BT system. However for LocalBT leechers, there are competitions among them and they just download data within local regions, which limits their gain of service capacity from foreign regions. The proof is therefore complete.

The main findings of this model are:

- Compared to increasing the number of freeriders beyond the local region of the client, increasing the number of freeriders within its local region will more intensely degrade the downloading rate of the LocalBT client. Freeriders are more harmful to LocalBT systems than native BT systems.
- Decreasing the ratio of LocalBT leechers to native BT leechers within local region will slightly improve the downloading rate of LocalBT leechers. Contrarily, increasing the number of native BT leechers beyond the local region will degrade the downloading rate of LocalBT leechers.
- Simplex local neighbors-selecting strategy is unfeasible, even though it has been widely deployed. Widely deploying LocalBT is not panacea. It can obviously improve the downloading performance by increasing LocalBT seeds and leechers in local regions. However, according to the Zipf-like geographical distribution of peers in the real BT systems, if there are too few peers within the LocalBT clients' local region, there will be poor downloading performance. In this situation, it can alternatively extend its local region to find more peers.

4 Performance Evaluation

A series of experiments were made on PlanetLab [19] to validate the model proposed in Section 3. In this section, the effects of freeriders, seeds, leechers and their geographical distribution on the performance of LocalBT clients were studied respectively. The details are described as below.

- Deploying clients. We exploited Vuze [20] as original BT clients, and developed plugins on Vuze as LocalBT clients, i.e., *ISP-bias*, *city-bias* and *country-bias* BT clients. Just as its name implied, an *ISP-bias* client just chose all its neighbors from its local ISP. The ISP information of a peer can be achieved by using the IP Prefix-to-AS mapping service of iPlane [21]. While, a *city-bias* BTclient or a *country-bias* BT client just chooses all its neighbors from its own city or country. The city and country information of a peer can be obtained by using the Geolite City/Country database [22], which could provide high accuracy of mapping IP-to-geolocation services.
- Publishing a torrent. One tracker server was deployed and a torrent was published about our teaching video, which is 83MB and with no copyrights issues.
- Choosing Hosts. Since the available capacities of most PlanetLab nodes are very high and it is very difficult to accurately measure available bandwidth between PlanetLab nodes, 180 PlanetLab nodes belonging to different ASes, cities, and countries were randomly chosen.

4.1 Impact of Freeriders and Seeds

To evaluate the impacts of freeriders and seeds on the downloading rate of LocalBT clients, 8 sets of experiments were conducted. Therein 4 of them were made for evaluating freeriders, i.e., scenario1 (S1), scenario2 (S2), and scenario 3 (S3 and S3+). Another 4 sets of experiments were made for evaluating seeds, i.e., scenario 4

(S4), scenario5 (S5), scenario6 (S6), and scenario 7 (S7). Table 1 lists the detailed parameters in the experiments. The tested clients in S1-S7 exploited *ISP-bias* BT protocol, while the test client in S3+ exploited native BT protocol.

Table 1. Parameters in four experiments for freeriders and four experiments for seeds

	S1	S2	S3	S3+	S4	S5	S6	S7
$N_{iR}^{freeRider}$	*0*	*0*	*5*	*5*	2	2	2	2
$N^{freeRider}$	*0*	*5*	*5*	*5*	10	10	10	10
$N_{iR}^{\bar{L}seed}$	2	2	2	2	*1*	*1*	*5*	*1*
N_{iR}^{Lseed}	3	3	3	3	*1*	*5*	*1*	*1*
N^{seed}	5	5	5	5	*10*	*14*	*14*	*16*
$N_{iR}^{Lleecher}$	10	10	10	10	20	20	20	20
$N_{iR}^{\bar{L}leecher}$	10	10	10	10	20	20	20	20
$N_{i\bar{R}}^{Lleecher}$	10	10	10	10	25	25	25	25
$N_{i\bar{R}}^{\bar{L}leecher}$	10	10	10	10	25	25	25	25
Total number	45	50	50	50	110	114	114	116

Fig. 3 illustrates that both LocalBT clients and native BT clients will get slower downloading rate when increasing the number of freeriders in the swarm. Compared to native BT, LocalBT clients' downloading rate will degrade more drastically when increasing the number of local freeriders. This maybe because local freeriders potentially consume more of the limited opportunities of LocalBT clients being unchoked by local seeds and other local leechers. This measurement study shows that freeriders will do more harm to the downloading rate of LocalBT clients than native BT clients. So it is necessary to design effective incentive mechanisms encouraging peers to upload more data in LocalBT systems.

Fig. 4 shows that LocalBT clients will more or less get faster downloading rate when increasing the number of seeds in the swarm. Compared to increasing the number of native BT seeds, increasing the number of LocalBT seeds will improve the downloading rates of LocalBT clients remarkably. This maybe because LocalBT seeds will provide more unchoking opportunities to LocalBT clients. Increasing the number of seeds beyond the client's local region will contribute limited capacities to LocalBT clients since we assume that the LocalBT clients just exchange data with local peers.

It is worth that LocalBT clients' downloading rate almost depends on the geographical distribution of seeds and freeriders to a high degree. There are more LocalBT seeds or less freeriders within their local region, or LocalBT clients will get slower downloading rate.

We also exploited *country-bias* BT clients instead of *ISP-bias* clients on the above scenarios. Since they perform as well as ISP-bias clients, we do not show the results in our paper.

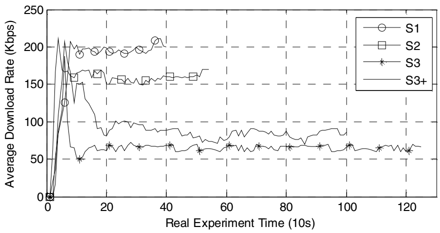


Fig. 3. Comparision of downloading rate with different numbers of freeriders

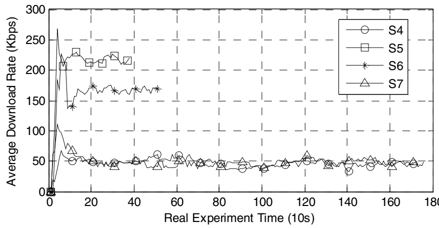


Fig. 4. Comparision of downloading rate with different numbers of seeds

4.2 Impact of Leechers

We also evaluate the impacts of leechers on the downloading rate of LocalBT clients. 5 sets of experiments were conducted, i.e., scenario8 (S8), scenario9 (S9), scenario 10 (S10), scenario 11 (S11) and scenario 12 (S12). The tested clients in S8-S12 exploited *ISP-bias* BT protocol. Table 2 lists the detailed parameters in the experiments.

Figure 5 illustrates that the average downloading rate will become slightly slower when decreasing the values of φ or increasing the values of λ. Decreasing the values of φ is equal to increasing the number of LocalBT leechers within local region, which may increase the competitions among them. The downloading rate as a function of φ is presented in S8-S10. We change the values of λ in S10-S12. Increasing the values of λ is equal to increasing the number of native BT leechers beyond the local region, which may increase the competitions between LocalBT leechers and native BT leechers. That is because native BT leechers beyond the local region may consume the limited resources of LocalBT leechers. Besides competitions, there are indirect reciprocities among BT peers, i.e., foreign native BT leechers help the local native BT leechers who may help local LocalBT leechers at the same time. So the average downloading rates in such scenarios do not present dramatically changes.

4.3 Impact of Geographical Distribution of Peers

We studied the impacts of geographical distribution of peers on downloading rate of LocalBT clients. We say that a node is on local sparse mode with respect to there are very few peers within its local neighborhood. Contrarily the node is in a dense mode. We conducted 4 sets of experiments. Two of them are sparse scenarios, i.e., scenario 13 (S13) and scenario 14 (S14). Another two of them are dense scenarios, i.e., scenario 15 (S15) and scenario 16 (S16). In S13 and S14, the tested clients exploited the *ISP-bias* BT protocol and *city-bias* BT protocol respectively. In S15 and S16, the tested clients exploited the *country-bias* BT protocol and native BT protocol respectively. S13, S14, S15 and S16 are the same scenario but with different views. For native BT clients, all peers in the swarm can be seen in their local region (e.g. S16). Since a *country-bias* BT client takes the country as its local region, which maybe larger than that view of a *city-bias* BT client or an *ISP-bias* BT client, Table 2 shows the detailed differences.

Table 2. Parameters in experiments for leechers and geographical distribution of peers

	S8	S9	S10	S11	S12	S13	S14	S15	S16
$N_{iR}^{freeRider}$	5	5	5	5	5	0	2	6	10
$N^{freeRider}$	15	15	15	15	15	10	10	10	10
$N_{iR}^{\bar{L}seed}$	10	10	10	10	10	0	5	10	15
N_{iR}^{Lseed}	10	10	10	10	10	0	0	5	5
N^{seed}	20	20	20	20	20	20	20	20	20
$N_{iR}^{Lleecher}$	5	10	20	20	20	5	5	10	15
$N_{iR}^{\bar{L}leecher}$	20	20	20	20	20	5	10	20	30
$\varphi = N_{iR}^{\bar{L}leecher} \big/ N_{iR}^{Lleecher}$	*4*	*2*	*1*	*1*	*1*	1	2	2	2
$N_{i\bar{R}}^{Lleecher}$	15	15	15	25	30	10	10	5	0
$N_{i\bar{R}}^{\bar{L}leecher}$	15	15	15	25	30	25	20	10	0
$\lambda = N_{i\bar{R}}^{\bar{L}leecher} \big/ N_{iR}^{\bar{L}leecher}$	*3/4*	*3/4*	*3/4*	*5/4*	*6/4*	5	2	1/2	0
Total number	90	95	105	125	125	75	75	75	75

Figure 6 compares average downloading rates of *ISP-bias* BT client, *city-bias* BT client, *country-bias* BT client, and native BT client on the same scenario. When one client is in a sparse mode of its local region, LocalBT clients probably perform worse than native BT clients. If there are too few peers within the client's local region, the client should extend its local region by shifting from *ISP-bias* BT protocol to *city-bias* BT protocol, or *country-bias* BT protocol, even native BT protocol.

According to the Zipf-like geographical distribution of peers in real BT systems [12], proximal over 80 percent of peers who in a dense mode can more or less benefit from LocalBT protocols. For those less than 20 percent of peers, simplex local neighbors-selecting strategy is unfeasible. If there are too few peers within their local regions, the best strategy for them maybe native BT protocol.

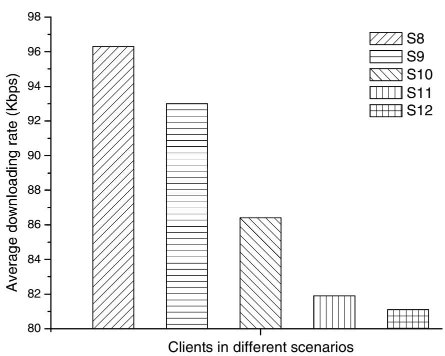


Fig. 5. Comparisions of downloading rate with different numbers of leechers

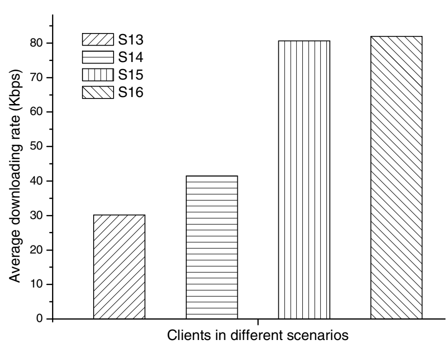


Fig. 6. Comparisions of downloading rate with geographical distribution of peers

5 Conclusion and Future Work

In this paper, a general analytical framework was proposed to quantify downloading performance of a large family of locality-aware BitTorrent protocols, i.e., *ISP-bias* BT protocol, *city-bias* BT protocol and *country-bias* BT protocol. We find that the downloading performance of LocalBT clients depends on the geographical distribution of peers to a high degree. When one client is in a sparse mode of its local region, LocalBT clients probably perform worse than native BT clients. If there are too few peers within the client's local region, the client should extend its local region by shifting from *ISP-bias* BT protocol to *city-bias* BT protocol, or *country-bias* BT protocol, even native BT protocol. For proximal less than 20 percent of peers in the swarm, simplex local neighbors-selecting strategy is unfeasible.

A series of experiments were made on PlanetLab to validate the framework. Results show that seeds, freeriders and leechers have different effects on the downloading performance of LocalBT clients. Increasing the number of seeds will remarkably improve the downloading performance of LocalBT clients. Different from previous researches [18], we find that increasing the number of native BT leechers beyond the local region will slightly degrade the downloading rate of LocalBT clients. For LocalBT clients, there are competitions among them and they just download data within local regions, which limits their gain of service capacity from other peers beyond local regions. We also find that freeriders are more harmful to LocalBT systems than native BT systems. Nevertheless, a more subtle model considering peer churn and a more effective incentive mechanism will be our future work.

Acknowledgments. Thanks for the anonymous reviewers and their valuable advice. This work was partly supported by the Natural Science Foundation of Jiangsu Province (No. BK2010133)

References

1. Cohen, B.: Incentives build robustness in Bittorrent. In: Proc. of P2PEcon, Berkeley, CA (June 2003)
2. Karagiannis, T., Rodriguez, P., Papagiannaki, K.: Should internet service providers fear peer-assisted content distribution? In: Proc. of ACM IMC, Berkeley, CA (October 2005)
3. Bindal, R., Cao, P., Chan, W., Medved, J., Suwala, G., Bates, T., Zhang, A.: Improving traffic locality in BitTorrent via biased neighbor selecting. In: Proc. of IEEE ICDCS, Lisboa, Portugal (July 2006)
4. Aggarwal, V., Feldmann, A., Scheideler, C.: Can ISPs and P2P systems cooperate for improved performance? ACM SIGCOMM Computer Communications Review 37(3), 29–40 (2007)
5. Xie, H., Yang, Y.R., Krishnamurthy, A., Liu, Y., Siberschatz, A.: P4P: Provider portal for applications. ACM SIGCOMM Computer Communications Review 38(3), 351–362 (2008)
6. Choffnes, D., Bustamante, F.: Taming the torrent: A practical approach to reducing cross-ISP traffic in peer-to-peer systems. ACM SIGCOMM Computer Communications Review 38(3), 363–374 (2008)

7. Agarwal, S., Lorch, J.: Matchmaking for online games and other latency-sensitive P2P systems. ACM SIGCOMM Computer Communications Review 39(3), 315–326 (2009)
8. Wang, H., Liu, J., Xu, K.: On the locality of BitTorrent-based video file swarming. In: Proc. of IEEE IPTPS, Boston, MA (April 2009)
9. Ren, S., Tan, E., Chen, S., Guo, L., Zhang, X.: TopBT: A topology-aware and infrastructure-independent BitTorrent client. In: Proc. of IEEE INFOCOM, San Diego, CA (March 2010)
10. Piatek, M., Madhyastha, H., John, J., Krishnamurthy, A., Anderson, T.: Pitfalls for ISP-friendly P2P design. In: Proc. of HOTNETSVIII, New York, USA (October 2009)
11. Lehrieder, F., Oechsner, S., Hoßfeld, T., Despotovic, Z., Kellerer, W., Michel, M.: Can P2P-users benefit from locality-aware? In: Proc. of IEEE P2P, Delft, Netherlands (August 2010)
12. Yu, L., Chen, M.: Geographical distribution of peers of bittorrent in multi-granularity scale. In: Proc. of ICCCT, Allahabad, India (September 2010)
13. Piatek, M., Isdal, T., Anderson, T., Krishnamurthy, A., Venkataramani, A.: Do incentives build robustness in BitTorrent? In: Proc. of USENIX NSDI, Cambridge, MA (April 2007)
14. Fan, B., Lui, J., Chiu, D.: The design trade-offs of bittorrent-like file sharing protocols. IEEE/ACM Transaction On Networking 17(2), 365–376 (2009)
15. Chow, A., Golubchik, L., Misra, V.: BitTorrent: An extensible heterogeneous model. In: Proc. of IEEE INFOCOM, Rio de Janeiro, Brazil (April 2009)
16. Qiu, D., Srikant, R.: Modeling and performance analysis of BitTorrent-like peer-to-peer networks. In: ACM SIGCOMM, Portland, USA (August 2004)
17. Legout, A., Urvoy-Keller, G., Michiardi, P.: Rarest first and choke algorithms are enough. In: Proc. ACM IMC, Rio de Janerio, Brazil (October 2006)
18. Guo, L., Chen, S., Xiao, Z., Tan, E., Ding, X., Zhang, X.: Measurements, analysis, and modeling of BitTorrent-like systems. In: Proc. of ACM IMC, Berkeley, CA (October 2005)
19. PlanetLab, http://www.planet-lab.org
20. Azureus BitTorrent Client, http://www.azureus.com/
21. iPlane: An information plane for distributed service, http://iplane.cs.washington.edu/data/origin_as_mapping.txt
22. Maxmind, MaxMind GeoIP Country Database, http://www.maxmind.com/app/country

Practical and Effective Domain-Specific Function Unit Design for CGRA

Ming Yan[1,*], Ziyu Yang[1], Liu Yang[2], Lei Liu[1], and Sikun Li[1]

[1] School of Computer, National University of Defense Technology
Changsha, Hunan, China. 410073
{mingyan,zyyang,liulei}@nudt.edu.cn, lisikun@263.net.cn

[2] College of Computer and Communication, Hunan University
Changsha, Hunan, China, 410082
yangliucn@126.com

Abstract. Coarse-grained reconfigurable architecture (CGRA) brings more powerful performance than the traditional CPU-like architecture. In this paper, we proposed a fast and effective domain-specific design method for function units of a CGRA named ProDFA. The proposed design method mainly concludes a top-down subgraph enumeration algorithm and a heuristic subgraph identification process based on topological searching. We used a clustering technique to accelerate the maximal valid subgraph enumeration (MVSE), and for the first time the top-down MVSE is combined with the identification process through the topological ordering. Candidate convex subgraphs are finally grouped into function sets according to their isomorphism. Experimental result shows that the performance of subgraph enumeration is improved in most cases, and a small number of candidate function sets are identified.

Keywords: Coarse-grained reconfigurable architecture, domain-specific design, function unit design, maximal valid subgraph enumeration, subgraph identification.

1 Introduction

Reconfigurable computing architecture is considered to be efficient and power saving for high performance computing in embedded systems. Many novel coarse grained reconfigurable architectures (CGRA) have been proposed during the last decade, such as MorphSys[1], PipeRench[2], ADRES[3], RICA[4]etc. Reconfigurable computing architecture promises attractive gain in both performance and efficiency for domain applications, but the domain-specific design is a hard and time-consuming job which concludes all the design aspects of architecture such as function unit design, architecture framework design, interconnection design and application mapping, etc. In order to design an efficient reconfigurable architecture for a certain application domain, much analytical work of application

* The work of this paper is financially supported by National Science Foundation of China with grant No. 61076020 and 90707003.

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 577–592, 2011.

codes and innovative designing of all aspects of hardware and software needs to be done. Design methodology and automatic tools are needed to aid the architecture designers.

Function unit design of reconfigurable architecture is very important, which decides the overall efficiency of the architecture. In this paper, based on a CGRA named ProDFA that we proposed, the design method of function unit is explored. The design methodology of CGRA function unit is much the same as instruction customization for application specific instruction processors(ASIP). Three phases are mainly concluded such as convex subgraph enumeration and candidate subgraph identification during architecture design, and custom instruction selection during application mapping. In this paper, we focus on the former two phases. Many state-of-the-art algorithms and frameworks have been proposed and developed for ASIP design. The work proposed in this paper is based on many great ideas from these algorithms.

In section 2, the background of related algorithms and design methods is summarized. And also the motivation of this paper is introduce with a brief introduction of ProDFA. A fast subgraph enumeration algorithm with a clustering technique and graph splitting process is then described in section 3, which is used to grab all the maximal valid subgraphs (MVSs) from the hot-spot application code. The heuristic searching process and subgraph grouping method are described in section 4. The experimental result of the MVS enumeration and subgraph identification is showed in section 5. And finally we give our conclusion in section 6.

2 Background and Motivation

2.1 Related Works

Atasu, et al. [6] proposed an algorithm to enumerate all valid subgraphs by travelling a binary tree constructed from a source dataflow graph (DFG) with topological order. This algorithm is an exact algorithm which could be integrated with identification and selection process [16]. Heuristic rules, such as IO port number, latency, and area, could be easily used during the search process for identification [7] and pruning the exploration space [8] . But this algorithm has a exponential complexity. While node number of the source DFG increases, these algorithms may become not feasible in reasonable execution time. Nagaraju, et al. [9] proposed a fast enumeration algorithm through enumerating the maximal convex subgraphs to identify candidate instructions with the absence of IO constraints. Paolo et al. [10] proposed a polynomial-time enumeration algorithm based on multiple vertex dominator and accelerated it with some heuristic pruning techniques in subgraph enumeration. Atasu, et al. [14] improved the tree travelling algorithm with a clustering technique which could reducing the complexity. Ajay, et al. [13] extended this clustering method with more pruning techniques and IO serialization method to identify custom-instructions. Lam, et al. [16] build a custom instruction design framework for reconfigurable instruction set processor (RISP) based on the exhaust search algorithm with some heuristic pruning rules. A pattern library and pattern templates are used to support domain applications exploration and hardware generation.

All the algorithms and methods referred above follow a bottom-up methodology which searches the DFG nodes one after another in a recursive way. Chen, et al. [11] proposed a unite and split algorithm to accelerate the identification process. The split idea using invalid nodes is extended to a top-down methodology for MVS enumeration by Li, et al. [15]. But the DnC algorithm [15] proposed by Li, et al. could not be used for identification directly. Topological order should be introduced so as to be able to search and identify after the MVS enumeration.

And though the top-down MVS enumeration algorithm proposed in [15] is much faster than bottom-up search algorithms, it could still be improved with some techniques. In this paper, a cluster technique is used to reduce the invalid node number, then an invalid node elimination and split algorithm is used for MVS enumeration. Since the top-down method in [15] can not be used for identification directly, topological order of the DFG is introduced right after the cluster operation. After that a search and partition algorithm with heuristic rules is then used on MVSs to identify candidate functions. Finally, candidate functions is grouped into candidate function sets according to their isomorphism and guide function values to aid the reconfigurable function unit design.

2.2 Introduction of ProDFA

The programmable dataflow computing architecture (ProDFA) is a coarse-grained reconfigurable architecture with highly flexible programmability. The overall structure of ProDFA is showed as Fig. 1. There are four sub-figures in Fig. 1. The sub-figure A shows the micro-architecture of the reconfigurable processing unit (RPU) of ProDFA. The RPU has a reconfigurable datapath pipeline which consists of several configurable function units and local memory. The sub-figure B shows the structure of the configurable application-specific function unit (CAFU). The sub-figure C is the structure of a reconfigurable processing node(RPN) which has four RPUs clustered. And the sub-figure D is an array structure composed of RPNs. With a hierarchical structure, the ProDFA could be expanded to large scale when the application requires for more powerful performance.

The programmability and efficiency is maintained through the RPU which is the basic element of ProDFA. There are mainly three parts inside a RPU showed as sub-figure A: the control and configure part showed inside the left blue dots, the execution part inside the center green dots, and the expandable interface part in the right red dot circle. ***The control and configure part*** is composed of a configware memory (CRAM), a runtime control Finite-State-Mechine (FSM), a control-status-register (CSR) and an execution context buffer. The control part supports every-cycle-control of the execution with much simpler control logic compared to VLIW or EPIC architecture. ***The execution part*** is the execution engine of the RPU, which concludes several configurable application specific function units (CAFU), local data memory (LMEM), configuration and interconnection bus (CIB) and a synchronization unit (SYNC) used to synchronize

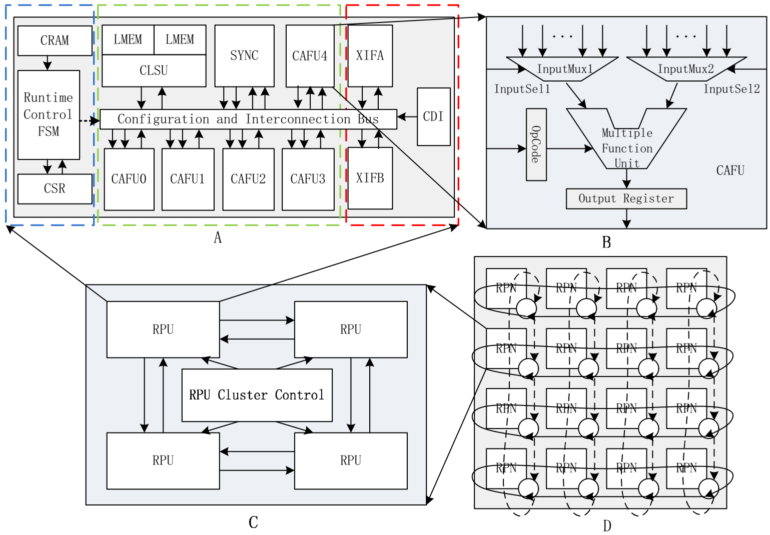


Fig. 1. The overall architecture of ProDFA, A: the architecture of the ProDFA reconfigurable processing unit (RPU); B: the structure of the configurable function unit (CAFU); C: the structure of the ProDFA reconfigurable processing node (RPN); D: the array structure of ProDFA;

the dataflow inside the RPU. The CAFUs are specially designed according to domain-specific applications, and each has a configure port which could register the function and the input data connection information issued from the runtime control FSM. The structure of CAFU is showed as Fig.1 part B. The input of CAFU is selected from all the outputs of other CAFUs. The input select information and opcode of each CAFU are issued from the runtime control unit and registered in each CAFU. ***The expandable interface part*** is composed of three input-output interfaces: external interface A (XIFA), external interface B (XIFB) and cluster data input (CDI). The external interface A and B are two pairs of data input output channels which connect one RPU to its neighbors using a port mapping scheme. The data comes from XIF is treated as a selectable input operand to all CAFUs. Cluster data input (CDI) is a special data input interface which supports data broadcasting when RPUs are clustered. The data comes from CDI is connected to one CAFUs which support a dedicated input operand.

2.3 Domain-Specific Design Flow

Since ProDFA defines the overall architecture which include the cluster structure, the datapath pipeline inside the RPU and the execution model of RPU, the function units is left to be designed for a given application domain. The domain-specific design flow is showed as Fig.2.

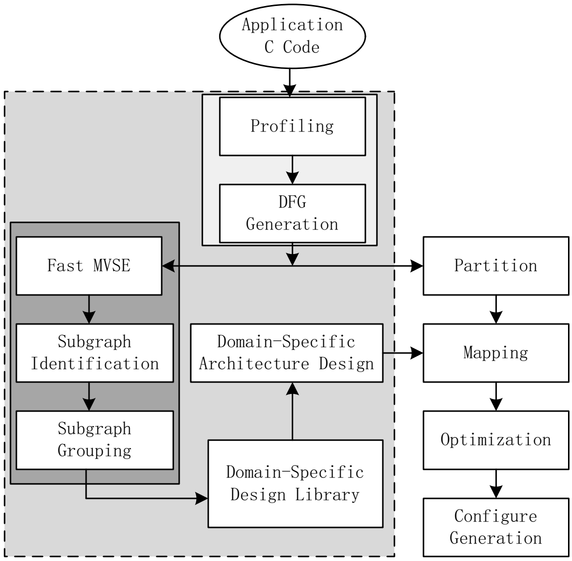


Fig. 2. Domain-Specific Design Flow of ProDFA

The domain-specific architecture design flow is showed in the light gray circle. The design flow outside the circle is the compiling flow which is not discussed in this paper. We used Trimaran as a front end tool which is used to profiling the application code and generating the dataflow graph (DFG) of hotspot code region. After the DFGs are obtained, three processes described in this paper are applied to these DFGs. The candidate function sets are then listed in a domain-specific design library. The library provides domain-specific design information to architecture designer. The domain-specific architecture design is done by hand for now which is being developed to automatic synthesis. After the specific function units are designed, an application mapping process will be applied to generate configuration of ProDFA.

3 Fast MVS Enumeration

The fast maximal valid subgraph enumeration algorithm is based on the directed acyclic dataflow graphs (DFG) generated from basic blocks of application code hot-spots. Given a DFG $G(V, E)$ and node $u \in V$, we use definitions listed as follow:

1. $Pred(G, u) = \{v | v \in V, v \neq u$, there is a path in G from v to $u\}$: Predecessors of u.
2. $Succ(G, u) = \{v | v \in V, v \neq u$, there is a path in G from u to $v\}$: Successors of u.
3. $Disc(G, u) = \{v | v \in V, v \neq u$, there is neither a path from u to v nor from v to $u\}$: Disconnected Set of u.

4. $iPred(G, u) = \{v|v \in Pred(G, u), v \neq u, u$ is adjacent from $v\}$:Immediate Predecessors of u.
5. $iSucc(G, u) = \{v|v \in Succ(G, u), v \neq u, u$ is adjacent to $v\}$: Immediate Successor of u.

3.1 Clustering Invalid Nodes

The basic idea of the enumeration algorithm is to split the original DFG into MVSs through eliminating the invalid nodes [15]. We assume that the original DFG of application code block is convex. In order to accelerate the splitting process, we first apply a clustering step to reduce the number of invalid nodes which decides the complexity of splitting. This cluster method is also adopted by other algorithms such as [14]. Unlike the algorithm proposed in [14] in which all nodes are clustered to reduce the search space for identification, we mainly focus on invalid nodes. In [14], the clustering will result in larger valid nodes which are clustered from several original nodes. This will make the searching process a little different because detailed information of the valid node is useful during a heuristic identification. Since invalid nodes are not considered during the identification searching process, the clustering of them will not change the process or any results of identification.

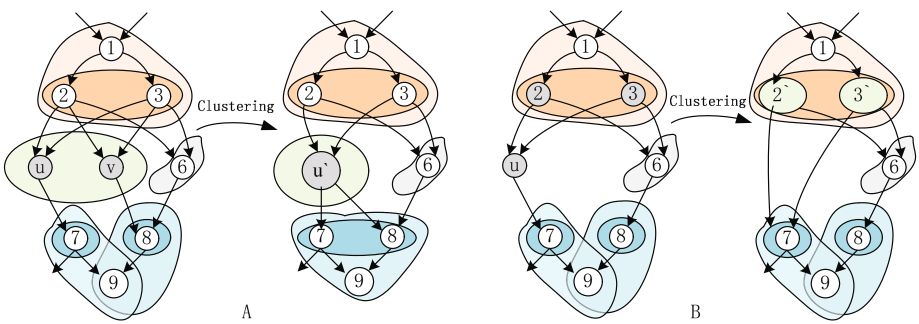


Fig. 3. Clustering the invalid nodes of the DFG

We define V_F as the set of invalid nodes in V of $G(V, E)$. For each $v \in V_F$, $iPred(v)$ and $iSucc(v)$ are checked with all the other invalid nodes. All nodes clustered to a same group are substituted by a new invalid node. G is transformed into G' and V_F is substituted by V'_F. In particular for nodes $u, v \in V_F$, the cluster operation is described as follows:

1. If $iPred(G, u) = iPred(G, v)$ **or** $iSucc(G, u) = iSucc(G, v)$, then nodes u and v are clustered together.
2. If $iPred(G, u) = \phi$ **or** $iSucc(G, u) = \phi$, then nodes u are eliminated directly as a boundray nodes.

3. If $iPred(G, u) \subseteq V_F$ **or** $iSucc(G, u) \subseteq V_F$, then nodes u could be clustered with nodes in $iPred(G, u)$ or $iSucc(G, u)$.

We prove the first statements after the split process is described because the proof needs a calculation based on the split operation. The second statement is proved in [15]. The proof of the third statement is showed as follows. Because the situation for $iPred()$ and $iSucc()$ are the same. We take $iPred()$ for example and prove that the subgraphs generated with or without cluster operation are equivalent. Because the subgraphs generated by splitting are equivalent with different order which invalid nodes are eliminated, we assume that node u is the last invalid node to be eliminated. Since node u has only invalid nodes as its $iPred()$, node u would be a boundary node in all the subgraphs after all its $iPred()$ are eliminated. So node u could be eliminated directly as boundary nodes which could be seen as the node u is clustered with all the nodes in $iPred()$. The cluster method is better explained by an example showed in Fig.3.

3.2 Eliminating V_F and Split

The basic idea of invalid nodes elimination process and graph splitting is much the same as algorithm proposed in [15] but with a slightly modification with the splitting process. After clustering process, the original DFGs of application code blocks are reduced in size and some invalid nodes have been substitute by new ones. In order to be able to search among the MVSs, the transformed DFG is topologically ordered before the splitting process.The elimination is a

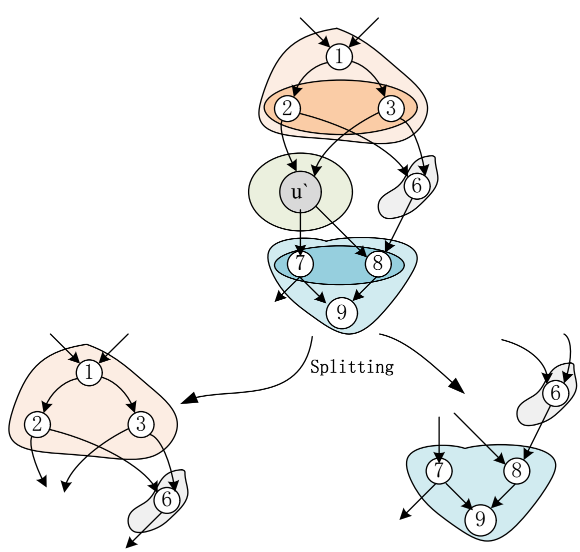


Fig. 4. Eliminating the invalid nodes and splitting of the DFG

recursion process which mainly consists of three parts: splitting, boundary node elimination and recursion.

1. If invalid node u is boundary nodes, which means $Pred(G, u) = \emptyset$ or $Succ(G, u) = \emptyset$, then node u could be simply eliminated from G.
2. If invalid node u is inside graph G, then the graph G splits into two subgraphs: G-$(Succ(G, u) \cup u)$=$Pred(G, u) \cup Disc(G, u)$ and G-$(Pred(G, u) \cup u)$=$Succ(G, u) \cup Disc(G, u)$.
3. The recursive process of elimination is applied to these two subgraphs separately.

The split process could be better explained by the example showed in Fig.4. The correctness of the splitting is proved in [15] and [11]. With the statements listed above, we now prove the correctness of the cluster operation.

Because the situation for $iPred()$ and $iSucc()$ is much the same. We take $iPred()$ for example. For $u, v \in V_F$ and $iPred(u) = iPred(v)$, obviously $Pred(u) = Pred(v)$. After cluster operation, the two new graph generated by splitting are: $sG1 = G - Pred() - \{u, v\}$ which equals to $Succ(u) \cup Disc(u) - \{v\}$ and $sG2 = G - Succ(u) - Succ(v) - \{u, v\}$. Without clustering, the split operation will generate three subgraphs in two steps. First step, eliminating the node u will generate $G1 = G - Pred() - \{u\}$ and $G2 = G - Succ(u) - \{u\}$. Second step eliminating the node v. Because v is a boundary node in $G1$, it could be eliminated directly and generate the $sG1$. When eliminating v in $G2$, two subgraphs will generated, one subgraph is $sG2$ and the other is $G - Succ(u) - \{u\} - Pred(v) - \{v\}$ which equals to $Disc(u) - \{v\}$. And $Disc(u) - \{v\}$ is a subgraph of $sG1$ which could be removed as a redundant graph. ∇

It is obvious that the two new subgraphs have an intersection $Disc(G, u)$ and MVSs in this part of subgraph will be enumerated more than once. The DnC algorithm [15] uses a redundancy check during the recursion which increases the execution time and makes the algorithm more complicated and difficult to calculate the complexity of it. The checking method is more like a heuristic rule during the splitting process. Because the graph splitting operation is simple based on set operation with bit-vector support, the redundancy is ignored in our algorithm. The worst case is when all the invalid nodes have no paths between each other and they are not boundary nodes in the new generated DFGs. The complexity of splitting process is $O(2^n)$ (n is the number of invalid nodes). Our experiment results show that the time cost increased by the number of splitting operation is smaller than the cost of checking process in most practical cases. Since the search process of identification is based on MVSs with topological order, the redundant MVSs could be easily removed later.

3.3 Enumeration Algorithm

The pseudo code description of the fast MVS enumeration algorithm is showed as Algorithm.1. The fast MVS enumeration algorithm mainly consists of four parts showed as Algorithm 1. First some preprocessing with the original DFG is done to prepare for clustering and splitting, such as calculating the $Pred(G, u)$ $iPred(G, u)$ and $Succ(G, u)$ $iSucc(G, u)$ for every invalid nodes in V_F. And then a clustering procedure is applied to reduce the number of invalid nodes. Before the splitting procedure is applied, the topological order of the clustered graph G' is calculated. Topological order is important and necessary for the identification process to keep the subgraph convex during searching.

Algorithm 1. Fast MVSE Algorithm

Input: $G(V, E)$ - Given DFG ;
    V_F - invalid nodes in $G(V, E)$
Output: $MVSs$ - The set of maximal valid subgraphs;
begin
  $MVSs = \emptyset$; // Preprocessing
For every node $u \in V_F$, calculate $Pred(G, u)$ $iPred(G, u)$ and $Succ(G, u)$ $iSucc(G, u)$
  Cluster(G, V_F, G', V_F'); // Apply invalid nodes clustering on G
  Topologically sorting the nodes of G'. // Sort the nodes with a topological order
  Split(G', V_F', $MVSs$); // Eliminating invalid nodes and splitting
end
Procedure: Split(G, V_F, $MVSs$)
begin
if V_F=$\emptyset$ **then** Add G to $MVSs$ and return;
else
  Select one invalid node $u \in V_F$;
  $SG_a = G - (Succ(G, u) \cup u)$; //Top half subgraph split from G
  $V_F^a = V_F \cap SG_a$;
  Calculate the $Pred(G, u')$ $iPred(G, u')$ and $Succ(G, u')$ $iSucc(G, u')$ of each node $u' \in V_F^a$ with SG_a;
  Boundary invalid nodes removal of V_F^a in SG_a;
  $Split(SG_a, V_F^a, MVSs)$;
  $SG_b = G - (Pred(G, u) \cup u)$;//Bottom half subgraph split from G
  $V_F^b = V_F \cap SG_b$;
  Calculate the $Pred(G, u')$ $iPred(G, u')$ and $Succ(G, u')$ $iSucc(G, u')$ of each node $u' \in V_F^b$ with SG_b;
  Boundary invalid nodes removal of V_F^b in SG_b;
  $Split(SG_b, V_F^b, MVSs)$;
end if
end

Unlike the DnC algorithm proposed in [15], the topological order is used in our enumeration algorithm for the identification searching process. The DnC algorithm is mainly used to enumerate all MVSs and gives no consideration of any further processing. And a special subgraph checking process is applied to avoid redundant MVSs generated by the intersection of $Disc(G, u)$. With the support of topological order, we use a much simpler splitting process and keep the redundancy until the searching process begins.

4 Heuristic Subgraph Identification

Design constraints is essential for a given architecture framework such as instruction pipeline with limited on-chip memory resources, datapath pipelines with limited on-chip interconnection resources, etc. Reconfigurable architecture is usually composed of several reconfigurable datapath pipelines formed by function unit arrays. Design constraints of function unit could be IO ports, area cost and operation latency. The heuristic subgraph identification concludes mainly three steps: redundancy removal, candidate convex subgraphs searching and convex subgraphs identification. With convex subgraph searching, the MVSs are partitioned into small convex subgraphs which satisfy the heuristic rules. These small convex subgraphs are then grouped into candidate function sets according to their isomorphism.

Algorithm 2. Candidate Subgraph Identification

Input: $MVSs$ - Sets of MVSs with topological order
Output: $CCSs$ - Set of candidate convex subgraphs
begin
Remove redundant MVSs with topological order;
For each MVS $sG \in MVSs$ **do**
  Attach the recurrence of the basic block to sG;
  For $InputPortNum = 1$ **upto** $In\_Maximal$ **do**
   For $OutputPortNum = 1$ **upto** $Out\_Maximal$ **do**
    MVS_Search(0,0,sG);
    MVS_Search(1,0,sG);
end
Procedure: MVS_Search;
Input: node_include, node_num, subgraph
Output: $CCSs$ - Set of candidate convex subgraphs
begin
  if there is no more nodes in subgraphs then
    add cur_CCS to $CCSs$; return; **end if**
  if $node\_include == 0$ then disable $Succ(sG, node\_num)$;
  else
    check_constraints(); //Check heuristic rules
    calc_gain_value(cur_CCS);
    update_candidate(cur_CCS, $CCSs$); **end if**
  $node\_num = node\_num + 1$;
  MVS_Search(0, $node\_num$, sG);
  if the next node is not disabled then MVS_Search(1, $node\_num$, sG); **end if**
  if $node\_include == 0$ then enable $Succ(sG, node\_num - 1)$; **end if**
end

4.1 MVS Searching and Identification

The heuristic candidate subgraphs identification is described as Algorithm 2.

After the enumeration process, many sets of MVSs are enumerated and each set is generated from one given DFG with nodes topologically ordered. There are very likely some redundant MVSs are generated in the split processing. In order to abandon the redundant MVSs, the MVSs are sorted according to the topological order of their first node and the node number. The MVSs with the same first node and the same node number are checked for redundancy. The checking process is simply to compare each node of the MVSs topologically which is very fast.

The MVS searching and partitioning take place right after the redundancy removal. Each MVS is searched individually. The search process uses the topological order and travels through the whole subgraph using a binary tree search method [6]. The topological search process makes sure that the identified subgraphs are convex subgraphs which could be scheduled properly. During the search process, heuristic rules are applied such as IO port number and latency constraints. Each subgraph that satisfies the constraints are recorded. Area cost and recurrence information is also attached to each subgraph before they are put into a candidate convex subgraph set (CCSs). Candidate subgraphs are classified according to their IO port numbers and ordered according to their gain value calculated from latency, area, number of nodes and recurrence. Number of nodes within each category is calculated for every candidate subgraph.

4.2 Heuristic Rules

Number of IO ports: Because the interconnection resource of RCA is limited, the number of IO port should be constrained. Usually, data to the input port of a function unit is selected from the output ports of all the other function units from the former stage of pipeline. So the number of output ports should be constrained as less as possible to decrease the latency and complexity of selection circuit. In our experiment, the output port number is constrained to be two maximal as $Out\_Maximal$ and the input port number is constrained to be four maximal as $In\_Maximal$.

Latency: Latency of function unit is very important for pipeline. The latency decides the clock frequency that the pipeline could support. Since there is usually local memory inside the RCA and the memory access cycle is almost certain with a specified manufacture technology. We use twice of the memory access cycle as the maximal latency of functions units. We use a relative latency of each functions compared to MAC. Latency of each operation is synthesized by Synoposys Design Compiler based on 0.13um CMOS standard cell library.

Area: Area of function unit is also important which decides the scale of the RCA. In our design method, the area information is also used during the subgraph grouping. So we add a area value related to the area of 32bits ADD unit. The area is not constrained with a maximal value to pruning the search space but to calculate the heuristic function gain value.

Recurrence: Since our design method is based on domain-specific applications, the hot-spot basic blocks of each application should be sorted in a reasonable way. So we introduce the recurrence of basic blocks to represent the importance of each DFG. The MVSs within the same DFG have the same recurrence value. Recurrence is not a pruning constraint either. It is used to order the candidate functions. After the searching and partitioning process, small convex subgraphs are sorted according to their gain value calculated from their latency, area, recurrence and node numbers of each graph.

Only two parameters are used as heuristic rules during searching: number of IO ports and the latency. The other two parameters: area cost and recurrence are used to calculate the gain value of each candidate subgraph. During the search process, other information of the subgraphs is gathered too, such as number of nodes, types of operations, etc. These information is used to group candidate subgraphs.

4.3 Convex Subgraph Grouping

In order to help designing efficient function units which usually have multiple functions and could be configured, the candidate functions should be divided into groups. Since a multiple function unit could cover several sub-functions naturally, the subgraph grouping uses a graph covering method based on their isomorphism and the information gathered during the searching process. Subgraphs that have only one node are omitted for candidate and its gain value is added to the operation.

Isomorphism checking is applied between the candidate subgraphs. Since graph isomorphism checking is time consuming, we used the type of nodes and number of nodes to guide the checking process. Two candidate subgraphs are checked for isomorphism only if the type of nodes and number of each type of nodes of one subgraph is larger than the other one, and the two subgraphs are not being covered by any other subgraph. The checking process begins with the important subgraphs which have bigger gain values. Subgraphs with isomorphic structure are treated as one subgraph, the smaller subgraph will be flagged as being covered and all gain values are accumulated to the bigger subgraph. If a function unit could support this big subgraph, then it could support most of the subgraphs by simply add MUXs before each operation.

5 Experiment and Results

The algorithms proposed in this paper are implemented in C++ programming language based on Trimaran compiler framework. The DFG of each hot-spot code block is generated before scalar register allocation and only flow data dependency is considered.

Table 1. Characteristics of Benchmarks

DFG	Benchmark	Procedure	Rgn.No.	Size	Weight
1	blowfish (large input) (encode)	_main	108	82	461761
2		_main	129	50	395796
3		_main	182(x6)	65	67657
4		BF_encrypt	2	380	406466
5		BF_cfb64_encrypt	19	55	81189
6	rijndael (large input) (encode)	_main	88	66	380572
7		_main	165(x7)	48	25371
8		_encrypt	2	1248	202973
9	rijndael (large input) (decode)	_main	127	60	380572
10		_main	192(x7)	27	25731
11		_decrypt	2	1248	202973
12	3DES	_main	79	11	20000
13		_des3_crypt	2	2256	40000
14		_des_crypt	2	822	20000
15		_des_main_ks	4	157	120
16	SHA	_sha_update	11	140	5983
17		_sha_update	18	70	38367
18		_sha_update	21(x3)	171	11571
19		_sha_update	72(x7)	143	609
20	MD5	_md5_process	2	812	458993
21		_main	32	90	25704
22		_main	36	65	54161
23	RC4	_main	11	69	622500
24		_main	13	125	789000
25		_main	21	170	62000

5.1 Benchmarks and Informations

As a case study of domain-specific applications, several block ciphers are chosen as benchmarks from Mibench and encrypt benchmark of Trimaran. We used hyper-block as the region level. Memory access operations and branch or jump operations are treated as invalid nodes which could not be supported by RCA function units efficiently. Showed as Table.1, the characteristics of the hot-spot code blocks is listed, such as benchmark with parameters, name of the procedure, the region number (Rgn.No.), DFG size of each block (Size), and the recurrence weight of each block. Some of the code blocks with the larger recurrence weights are listed. Region numbers with a multiplication such as 182(x6) in row of DFG 3 stands for there are total six regions which have a isomorphic DFG. So the recurrence weight of each MVSs from this DFG will be multiplied with 6.

5.2 Subgraph Enumeration and Identification Results

The results of the MVS enumeration and candidate functions (CFs) identification is showed as table 2. The number of invalid nodes before and after cluster operation is given in col.2 and col.3, followed by the total number of MVS and

Table 2. Performance of our Fast MVSE Algorithm and CF Results

DFG	V_F	V_F'	MVSs	Runtime	DnC[15]	CFs
1	49	5	6	0.16	0.22	1
2	26	0	1	0.10	0.12	1
3	28	1	2	0.12	0.20	2
4	89	64	49	0.85	0.86	6
5	24	1	2	0.11	0.12	2
6	40	0	1	0.33	0.38	1
7	13	3	8	0.11	0.11	0
8	363	356	5776	1241	1282	38
9	40	0	1	0.10	0.12	1
10	7	2	4	0.01	0.01	1
11	363	356	5776	1241	1282	38
12	6	0	1	0.01	0.01	1
13	496	384	1200	359	384	17
14	176	128	400	125	138	17
15	5	0	1	0.01	0.01	15
16	130	0	1	2.12	3.02	1
17	29	0	1	0.32	0.33	2
18	24	0	1	0.11	0.14	8
19	130	0	1	0.33	0.42	1
20	88	0	1	2.75	2.67	21
21	32	0	1	0.37	0.32	4
22	25	0	1	0.26	0.21	1
23	21	0	1	0.25	0.20	1
24	50	7	8	0.58	0.60	4
25	71	24	25	1.12	1.10	2

runtime of the MVS enumeration. In order to compare the performance of the cluster and simplified splitting algorithm with [15], we implemented the DnC algorithm and showed the runtime. The runtime has limited to 0.01ms minimal.

Showed as table 2, most of the DFGs have only one MVS because the invalid nodes are usually boundary nodes or become boundary nodes after cluster operation. Because the cluster process checks the $iPred()$ and $iSucc()$ of all invalid nodes, the algorithm will take some more time when there many invalid nodes but few of them could be clustered. But with a simplified splitting process, the enumeration process is accelerated with a larger memory cost. The algorithm in this paper is faster than DnC[15] in most cases. After the redundancy removal and isomorphism covering, the number of CFs is showed in the last column. Because small candidate functions which only have one operation is excluded, and with the help of heuristic pruning rules, the number of identified CFs is small. Some most weighted candidate function sets are showed in the Fig.5 which could be useful for cryptographic architecture design.

Three group of candidate function sets are showed in Fig. 5 each in a subgraph column. Because the heuristic rule (i.e. number of IO ports) is set to maximal 4 input ports and 2 output ports, most of the candidate function sets are subgraphs

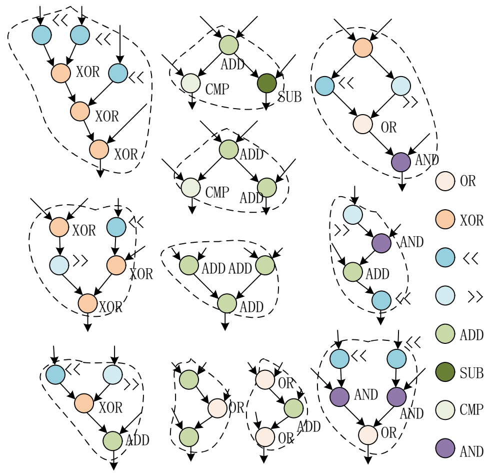


Fig. 5. Identified candidate function set examples for block ciphers

with 4 inputs. As a matter of fact, when we designed the specific function units (SFUs) of ProDFA for block ciphers, the CFs are used as prior options to be selected as functions supported by SFUs.

6 Conclusion

Top-down MVS enumeration (MVSE) algorithm is much faster than traditional bottom up enumeration methods in practical, because the invalid nodes are usually boundary nodes and the number of invalid nodes inside a DFG is usually smaller than valid nodes within big DFGs. To our knowledge, the work proposed in this paper is the first time a top-down MVSE algorithm is combined with a topological search process for function unit design of CGRA. The experimental result shows the improvement of MVSE performance and effectiveness of the identification and grouping method.

References

1. Lee, M.H., Singh, H., Lu, G.M., Bagherzadeh, N., et al.: Design and Implementation of the MorphoSys Reconfigurable Computing Processor. J. VLSI Signal Process. Syst, 147–164 (2000)
2. Goldstein, S.C., Schmit, H., Moe, M., Budiu, M., et al.: PipeRench: a co/processor for streaming multimedia acceleration. In: ISCA 1999: Proceedings of the 26th Annual International Symposium on Computer Architecture, pp. 28–39 (1999)
3. Mei, B., Vernalde, S., Verkest, D., Man, H.D., Lauwereins, R.: ADRES An Architecture with Tightly Coupled VLIW Processor and Coarse-Grained Recon?gurable Matrix. In: Y. K. Cheung, P., Constantinides, G.A. (eds.) FPL 2003. LNCS, vol. 2778, pp. 61–70. Springer, Heidelberg (2003)

4. Khawam, S., Nousias, I., Milward, M., Yi, Y., Muir, M., Arslan, T.: The Reconfigurable Instruction Cell Array. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 16(1), 75–85 (2008)
5. Kastner, R., Bozorgzadeh, E.: Instruction Generation for Hybrid Reconfigurable Systems. ACM Transactions on Design Automation of Electronic Systems 7(4), 605–627 (2002)
6. Atasu, K., Pozzi, L., Ienne, P.: Automatic Application-Specific Instruction-Set Extensions under Microarchitectural Constraints. In: DAC 2003, pp. 1–6 (2003)
7. Clark, N., Zhong, H., Mahlke, S.: Automated Custom Instruction Generation for Domain-Specific Processor Acceleration. IEEE Transactions on Computer 54(10), 1258–1271 (2005)
8. Pozzi, L., Atasu, K., Ienne, P.: Exact and Approximate Algorithms for the Extension of Embedded Processor Instruction Sets. IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems 25(7), 1209–1230 (2006)
9. Pothineni, N., Kumar, A., Paul, K.: Application Specific Datapath Extension with Distributed I/O Functional Units. In: 20th VLSI Design - 6th Embedded Systems, pp. 551–556 (2007)
10. Bonzini, P., Pozzi, L.: Polynomial-Time Subgraph Enumeration for Automated Instruction Set Extension. In: DATE 2007, pp. 1331–1336 (2007)
11. Chen, X., Maskell, D.L., Sun, Y.: Fast Identification of Custom Instructions for Extensible Processors. IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems 26(2), 359–368 (2007)
12. Atasu, K., Ozturan, C., Dundar, G., Mencer, O., Luk, W.: CHIPS: Custom hardware instruction processor synthesis. IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems 27(3), 528–541 (2008)
13. Verma, A.K., Brisk, P., Ienne, P.: Fast, quasi-optimal, and pipelined instruction-set extensions. In: Proceedings of the 2008 Asia and South Pacific Design Automation Conference, pp. 334–339 (2008)
14. Atasu, K., Mencer, O., Luk, W., Ozturan, C.: Fast Custom Instruction Identification by Convex Subgraph Enumeration. In: ASAP 2008, pp. 1–6 (2008)
15. Li, T., Sun, Z., Jigang, W., Lu, X.: Fast Enumeration of Maximal Valid Subgraphs for Custom-instruction Identification. In: CASES 2009, pp. 1–8 (2009)
16. Lam, S.-K., Srikanthan, T.: Rapid design of area-efficient custom instructions for reconfigurable embedded processing. Journal of Systems Architecture 55, 1–14 (2009)

Secure Hash-Based Password Authentication Protocol Using Smartcards

Hyunhee Jung[1] and Hyun Sung Kim[2,*]

[1] School of Computer Engineering, Kyungil University
712-701, Kyungsansi, Kyungpuk, Korea
gusgml7041@naver.com
[2] School of Computer Engineering, Kyungil University
712-701, Kyungsansi, Kyungpook Province, Korea
kim@kiu.ac.kr

Abstract. Recently, Jeong-Won-Kim proposed a hash-based strong-password authentication protocol and claimed that the protocol is secure against guessing attack, stolen-verifier attack, replay attack, and impersonation attack. However, we show that their protocol has two vulnerabilities, password guessing attack and authentication answer guessing attack. Furthermore, we present a secure hash-based password authentication protocol using smartcards to cope with the vulnerabilities. Security analysis shows that our protocol provides better security properties than the other related authentication protocols with the similar computational complexity with others.

1 Introduction

It is necessary to verify the identities of the communicating parties when they initiate a connection. Techniques for user authentication are broadly based on one or more of the following categories: (1) what a user knows, (2) what a user is, or (3) what a user has. Among them, the first category is the most widely used method due to the advantages of simplicity, convenience, adaptability, mobility, and less hardware requirement. It requires users only to remember their knowledge such as a password. However, traditional password-based protocols are susceptible to off-line password guessing attacks since users tend to choose easy memorable passwords with relatively low entropy [1-5].

Many researchers have proposed secure password-based schemes to cope with the problem [6-14]. In 2000, Sandirigama et al. proposed a simple and secure password authentication protocol but Lin et al. showed that the protocol was vulnerable to replay and denial-of-service (DoS) attacks [6-7]. Furthermore, they proposed an optimal strong password authentication scheme, which claimed that it is secure against all known attacks. Chen and Ku showed that Lin et al.'s protocol is vulnerable to stolen-verifier attack [8]. Lee et al. in [9] presented an improved authentication scheme to solve the vulnerabilities of password guessing attack in Peyravian et al.'s password

* Corresponding author.

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 593–606, 2011.

authentication scheme [10]. However, Yoon et al. showed that the Lee et al.'s scheme is vulnerable to DoS attack, and proposed improved protocol to solve the security problem [11]. Recently, Ku et al. demonstrated that Yoon et al.'s scheme is not secure against off-line password guessing attack and stolen-verifier attacks [12]. Moreover, Kim and Koç showed that Yoon et al.'s scheme is also vulnerable to DoS attack [13]. They also proposed a hash-based strong-password authentication protocol was described in [13]. But, Jeong-Won-Kim showed that the protocol in [13] is vulnerable to impersonation, guessing, and stolen-verifier attacks [14]. Furthermore, they proposed an improvement password authentication protocol to solve the problem in Kim and Koç's scheme.

This paper shows Jeong-Won-Kim's protocol still has two security weaknesses, password guessing attack and authentication answer guessing attack. To solve the problems in the previous protocols, we propose a secure hash-based password authentication protocol using smartcards. As we could see the security analysis, our protocol could not only provide better security properties than the other related authentication protocols but also provide similar computational complexities than the others.

Section 2 overviews the network architecture and reviews Jeong-Won-Kim's authentication protocol. In Section 3, we show the vulnerabilities of Jeong-Won-Kim's protocol. Sections 4 and 5 proposes a secure hash-based password authentication protocol using smartcards to solve the vulnerabilities of Jeong-Won-Kim's protocol and gives security and performance analyses. Finally, we conclude the paper in Section 6.

2 Related Works

This section reviews network architecture and the hash-based password authentication protocol proposed by Jeong-Won-Kim [14].

2.1 Network Architecture

This paper considers the client-server network architecture as shown in Fig. 1 [15]. The client–server model of computing is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients.

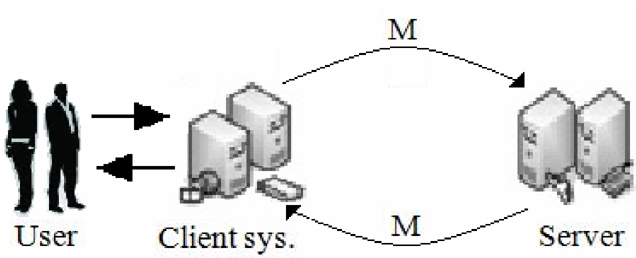


Fig. 1. Network Architecture

Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system. A server machine is a host that is running one or more server programs which share their resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await incoming requests.

2.2 Jeong-Won-Kim's Authentication Protocol

Jeong-Won-Kim's authentication protocol is consisted of four phases, registration, login, forget password, and password/verifier change [14]. We only review three phases including registration, login, and password/verifier change because they will be used for the cryptanalysis. Notations used in this paper is defined as follows

- U denotes the user and S denotes the server.
- U_{id} and ID denote the identification of the user.
- X_S denotes secret key of S.
- P denotes the password of U.
- K_u is a randomly generated key selected by U and shared with the server and stored in secure storage in a smartcard.
- r_n denotes a random nonce generated by U or S.
- T_s denotes timestamp.
- E_{Kpu} denotes a cryptographically secure public key encryption algorithm, such as RSA-OAEP(Optimal Asymmetric Encryption Padding) with the public key of S. It's used whenever user verifiers are stored in the registration phase.
- D_{Kpr} denotes RSA-OAEP decryption with the private key of S. It's used whenever user verifiers have to use in another phase, for example in the forget password phase and the password/verifier change phase.
- $E_{Spu}(M)$ denotes encryption of M with the public key of S when U sends M to S.
- $D_{Spr}(M)$ denotes decryption of M with the private key of S when U's $E_{Spu}(M)$ decrypts.
- $\oplus$ denotes the bitwise XOR operation, and $\|$ denotes concatenation.
- $Auth_Q$ and $Auth_A$ denote the authentication question and answer for the registration, forget password, and password/verifier change Phases.
- $h(\cdot)$ denotes a cryptographic hash function, such that $h(m)$ signifies the hash with message m. Furthermore, $h(a, b)$ denotes the hash of concatenated a and b; *i.e.*, $h(a, b) = h(a \| b)$.

[Registration Phase] In step R1, U computes password verifier $PV = h(K_u \| P) \oplus K_u$ and sends it to U. S computes $R = PV \oplus Ts$ with received PV and $Auth_Q$, and then sends it to U in step R2. U receives R2, and U computes the user's important verifier $UV = h(Ku \| P' \| Ts \| U_{id}) \oplus K_u$. Next, U encrypts UV, T_s', U_{id}, K_u, P, and $Auth_Q \oplus Auth_A$ with S's public key, and then sends it. After S receives R3, S decrypts R3 and computes $h(K_u \| P) \oplus K_u$. Then, S compares $h(K_u \| P) \oplus K_u$ and T_s' with PV and T_s that were stored in step R2. If both are equal, then S stores the sealed-verifier $SV = E_{Kpu}(h(K_u \| P \| U_{id}))$, PV, $UKP = E_{Kpu}(U_{id}, K_u, P)$, and $QAK = E_{Kpu}(Auth_Q \oplus Auth_A)$ in the S password file. Fig. 2 shows the phase.

U		S
	R1. $PV = h(K_u \,\|\, P) \oplus K_u$ $\longrightarrow$	
	$\longleftarrow$ R2. R, $Auth_Q$	
	R3. $E_{Spu}(UV, T_s', U_{id}, K_u, P, Auth_Q \oplus Auth_A)$ $\longrightarrow$	

Fig. 2. Registration Phase

[Login Phase] In step L1, U inputs his/her ID, password, and private key into the client system. The client system computes U's password verifier $PV' = h(K_u \,\|\, P) \oplus K_u$ and then sends it to S. U receives L1, and S compares PV' to PV stored in R2. If they are equal, then S generates a nonce r_s, and then sends PV and r_s to U in step L2. U receives L2, U compares PV' to PV. If they are equal, then U computes $L = h(h(K_u \,\|\, P \,\|\, U_{id}) \oplus PV \oplus r_s) \oplus h(K_u \,\|\, P \,\|\, U_{id}) \oplus PV \oplus r_s$, and sends it to S in step L3. S receives L3, S derives $C_1 = h(h(K_u \,\|\, P \,\|\, U_{id}) \oplus PV \oplus r_s) \oplus h(K_u \,\|\, P \,\|\, U_{id}) \oplus PV$ XORing L with r_s. S then computes $C_2 = SV \oplus K_{pr} = h(K_u \,\|\, P \,\|\, U_{id})$ and $C_3 = h(C_2 \oplus PV \oplus r_s) \oplus C_2 \oplus PV$ using the stored SV and K_{pr} in the S password file. Next, S checks $C_1 = C_3$. If they are equal, S authenticates U. Fig.3 shows the phase.

U		S
	L1. $PV' = h(K_u \,\|\, P) \oplus K_u$ $\longrightarrow$	
	$\longleftarrow$ R2. PV, r_s	
	L3. $L = h(h(K_u\|P\|U_{id}) \oplus PV \oplus r_s) \oplus h(K_u\|P\|U_{id}) \oplus PV \oplus r_s$ $\longrightarrow$	

Fig. 3. Login Phase

U		S
	PC1. Password-change request $\longrightarrow$	
	$\longleftarrow$ PC2. $Auth_Q$, r_s	
	PC3. $E_{Spu}(W, P_{new}, K_{unew}, U_{idnew})$ $\longrightarrow$	

Fig. 4. Password/verifier change Phase

[Password/Verifier Change Phase] First of all, U sends password-change request to S. Next, in step PC2, S generates a random nonce r_s, and sends $Auth_Q$ and r_s to U. U receives PC2, U computes $W = Auth_Q \oplus Auth_A \oplus r_s \oplus PV$, and encrypts W and the new values of K_{unew}, P_{new}, and U_{idnew} with S's public key. Next, U sends $E_{Kpu}(W, K_{unew},$

P_{new}, U_{idnew}) to S in step PC3. S receives PC3, S decrypts it, and obtains W, K_{unew}, P_{new}, and U_{idnew}. S then computes $W_2 = Auth_Q \oplus Auth_A$ using decrypted QAK with K_{pr}, which was stored in the S password file, and obtains PV in the file. Next, S computes $W_4 = W_2 \oplus r_s \oplus PV$ and checks $W = W_4$. If they are equal, then S stores the new $SV = E_{Kpu}(h(K_{unew} \parallel P_{new} \parallel U_{idnew}))$, new $PV = h(K_{unew} \parallel P_{new}) \oplus K_{unew}$, new $UKP = E_{K\text{-}pu}(U_{idnew}, K_{unew}, P_{new})$, and $QAK = E_{Kpu}(Auth_Q \oplus Auth_A)$ in the password file. Fig. 4 shows the phase.

3 Cryptanalysis of Jeong-Won-Kim's Protocol

This section shows that that the Jeong-Won-Kim protocol is vulnerable to two guessing attacks, password guessing attack and authentication answer guessing attack. First of all, we will show that the protocol is weak against to the password guessing attack in the login phase with the assumption that adversary could steal a user's smart card. After that, we will show that the protocol is vulnerable to the authentication answer guessing attack in the password/verifier change phase with the assumption that the attacker could perform stolen-verifier attack to the server.

3.1 Password Guessing Attack

Since the login request message from the legal user is sent to the server through an insecure channel in the login phase, we could assume that adversary can control the channel completely. After the adversary intercepts the login request message, and steals the smart card from the user, he/she can guess the password for the user by computing $PV' = h(K_u \| P') \oplus K_u$ and verifying it with the intercepted PV in the previous session messages by using the value K_u stored in the smart card. The overall procedure for the attack is as follows

1. An adversary intercepts messages from a legal user's communication session and steals the smart card from the user.
2. The adversary computes $PV' = h(K_u \| P') \oplus K_u$ with a guessed password P', where K_u is stored information on the smart card.
3. The adversary verifies his/her guess of the password P' by comparing PV' with the intercepted PV.

If the verification is successful, it means that the guessed password is the correct one. Otherwise, the adversary performs another try of the password guessing attack until he/she gets the matched password.

3.2 Authentication Answer Guessing Attack

A guessing attack for $Auth_A$ in the password/verifier change phase is possible with additional assumptions that the adversary could perform the stolen-verifier attack to the server and intercept $Auth_Q$ from one of the previous sessions among the forget password phase or the password/verifier change phase. By using the intercepted messages and the verifier from the server, the adversary could guess the correct authentication answer $Auth_A'$ by computing $QAK'=E_{Spu}(Auth_Q \oplus Auth_A')$ and verifying it with the stolen *QAK* from the server by using the encrypted message the intercepted value $Auth_Q$ and guessed $Auth_A'$ with the server's public key S_{pu}.

1. An adversary intercepts $Auth_Q$ from one of sessions among the forget password phase or the password/verifier change phase.
2. The adversary guesses the correct authentication answer $Auth_A'$.
3. The adversary computes $QAK' = E_{Spu}(Auth_Q \oplus Auth_A')$ by using the encrypted message the intercepted value $Auth_Q$ and guessed $Auth_A'$ with the server's public key S_{pu}.
4. The adversary compares QAK' with the stolen *QAK* from the server.
5. If they are equal, the adversary obtains the correct authentication answer $Auth_A$.

If the verification is successful, it means that the adversary's attack is successful. If the adversary obtains $Auth_A$, the adversary can also do impersonation attack. That is to say, the adversary impersonates the user with the stolen verifier after the following steps for the password/verifier change. Fig. 5 shows the password/verifier change phase for the impersonation and the detailed steps are as follows

1. An adversary guesses $Auth_A'$ using stolen $Auth_Q$.
2. The adversary computes $W' = Auth_Q \oplus Auth_A' \oplus \mathrm{rs} \oplus PV$ using stolen-verifier *PV* in Server's password file and defines K_{unew}, P_{new} and U_{idnew}. And then the adversary sends $E_{Spu}(W', K_{unew}, P_{new}, U_{idnew})$ to *S*.
3. *S* decrypts $E_{Kpu}(W', K_{unew}, P_{new}, U_{idnew})$, and obtains W', K_{unew}, P_{new}, and U_{idnew}. And then *S* computes $W_2 = Auth_Q \oplus Auth_A$ using decrypted *QAK* with K_{pr}, which was stored in *S*'s password file, and obtains *PV* in the file. Next, *S* computes $W_4 = W_2 \oplus r_s \oplus PV$ and checks $W' \mathrel{?=} W_4$. If they are equal, *S* stores the new $SV = E_{Kpu}(h(K_{unew} \| P_{new} \| U_{idnew}))$, new $PV = h(K_{unew} \| P_{new}) \oplus K_{unew}$, new $UKP = E_{Kpu}(U_{idnew}, K_{unew}, P_{new})$, and $QAK = E_{Kpu}(Auth_Q \oplus Auth_A)$ in the file.

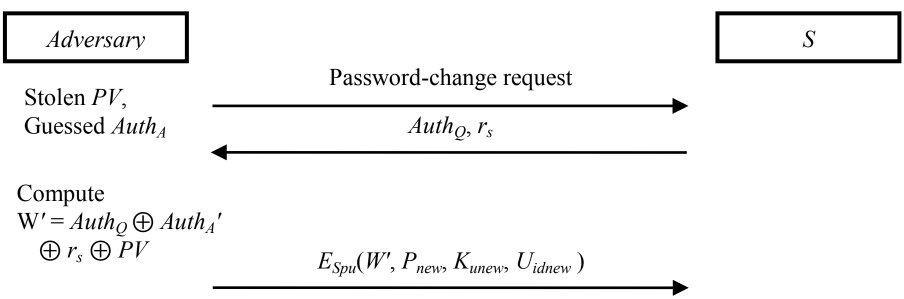


Fig. 5. Password/verifier change phase for the impersonation

4 Secure Hash-Based Password Authentication Protocol

This section proposes a secure hash-based password authentication protocol to solve the security problems in Jeong-Won-Kim's protocol. This protocol is consisted with registration, login, forget password and password/verifier change phases.

4.1 Registration Phase

This session supposes using secure channel. Fig. 6 shows the registration phase and the detailed steps are as follows

R1. $U \rightarrow S$: ID, PV

U inputs his/her ID and P, generates K_u, and computes the password verifier $PV=h(K_u||P)\oplus K_u$. U sends ID and PV to S for a registration request.

R2. $S \rightarrow U$: $Auth_Q$

S computes $IDX=E_{Kpu}(h(ID||X_S))\oplus X_S$ and stores it in S's password file. And S sends $Auth_Q$ to U.

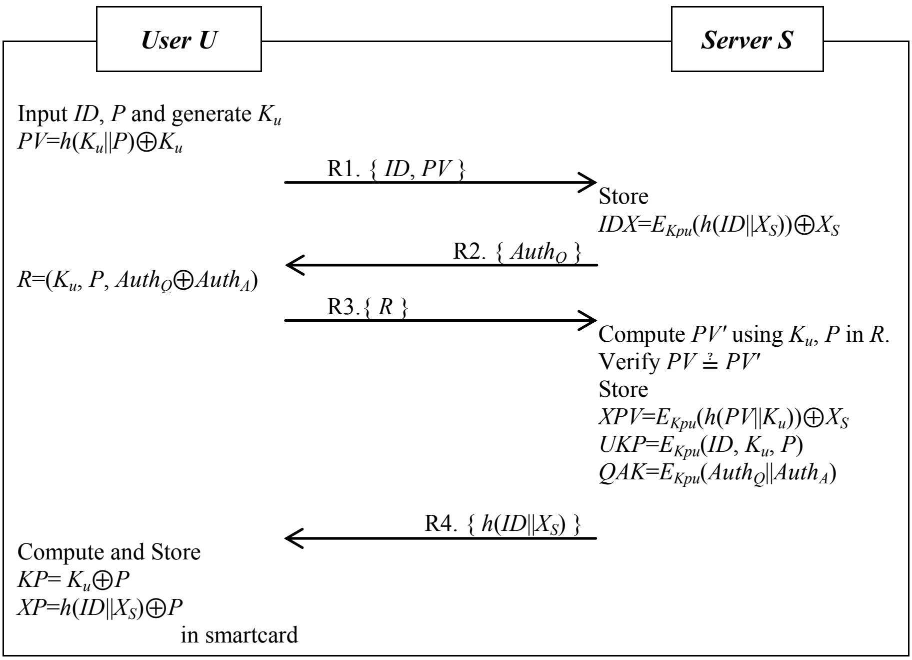


Fig. 6. Registration Phase

R3. $U \rightarrow S$: $R = E_{Spu}(K_u, P, Auth_Q\oplus Auth_A)$

U inputs $Auth_A$ as an answer for the authentication question $Auth_Q$ and computes $Auth_Q\oplus Auth_A$. Next, U encrypts K_u, P, $Auth_Q\oplus Auth_A$ with S_{pu} and sends it to S.

R4. $S \rightarrow U$: $h(ID||X_S)$

When S receives R3, S decrypts it and computes PV' using K_u and P from R. And S compares PV' with the received PV in R1. If they are equal, S stores $XPV=E_{Kpu}(h(PV||K_u))\oplus X_S$,

4.2 Login Phase

This phase uses the challenge-response method as protection from replay attack. Fig. 7 shows the login phase and the detailed steps are as follows

L1. $U \rightarrow S$: $E_{Spu}(XP \oplus P, ID, r_1)$

U enters his/her smartcard in the card reader, and inputs ID and P. Next, U generates a nonce r_1 and encrypts XP, r_1 and ID with S_{pu}. And then U sends it to S for a login request.

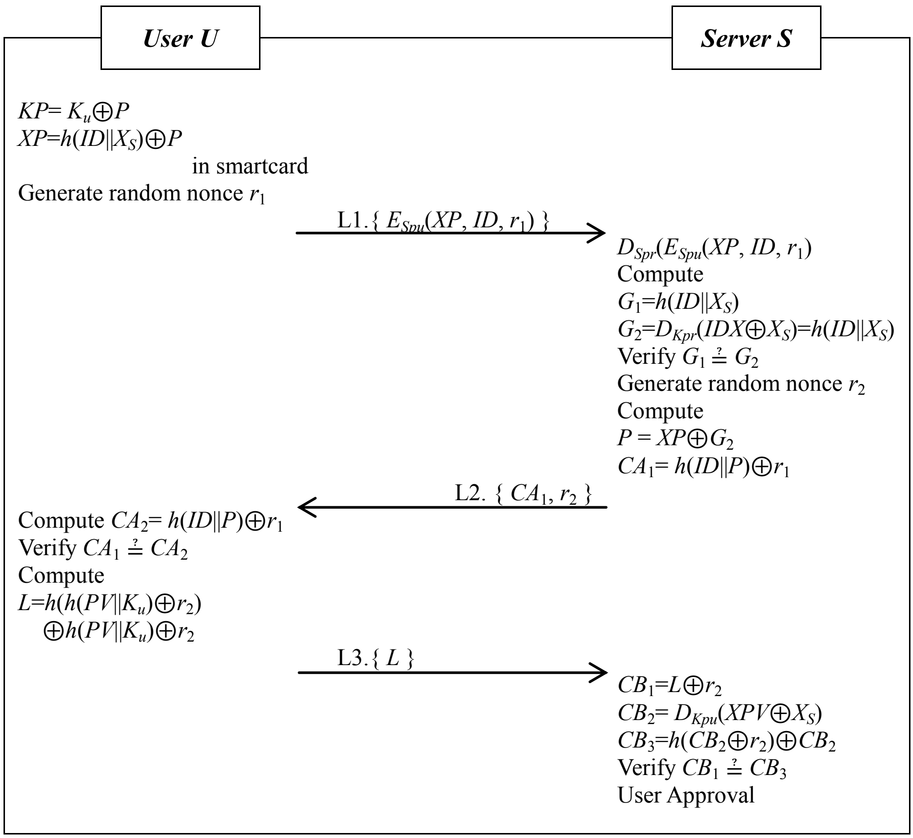


Fig. 7. Login Phase

L2. $S \rightarrow U$: CA_1, r_2

When S receives L1, S decrypts it and computes $G_1=h(ID||X_S)$ and $G_2=D_{Kpr}(IDX \oplus X_S) = h(ID||X_S)$ by decrypting $IDX \oplus X_S$ with S's private key K_{pr}. And S compares G_1 with G_2. If they are equal, S computes $P = XP \oplus G_2$ and $CA_1= h(ID||P) \oplus r_1$, generates a nonce r_2, and sends them to U.

L3. $U \rightarrow S$: $L = h(h(PV \| K_u) \oplus r_2) \oplus h(PV \| K_u) \oplus r_2$

U computes $CA_2 = h(ID \| P) \oplus r_1$ and compares CA_2 with the received CA_1. If they are equal, U computes $L = h(h(PV \| K_u) \oplus r_2) \oplus h(PV \| K_u) \oplus$c$r_2$, and sends L to S. When S receives L, S computes $CB_1 = L \oplus r_2$ and computes $CB_2 = D_{Kpr}(XPV \oplus X_S) = h(PV \| K_u)$ by decrypting $XPV \oplus X_S$ with S's private key K_{pr}. And then S computes $CB_3 = h(CB_2 \oplus r_2) \oplus CB_2$ and compares CB_3 with CB_1. If they are equal, U authenticates S.

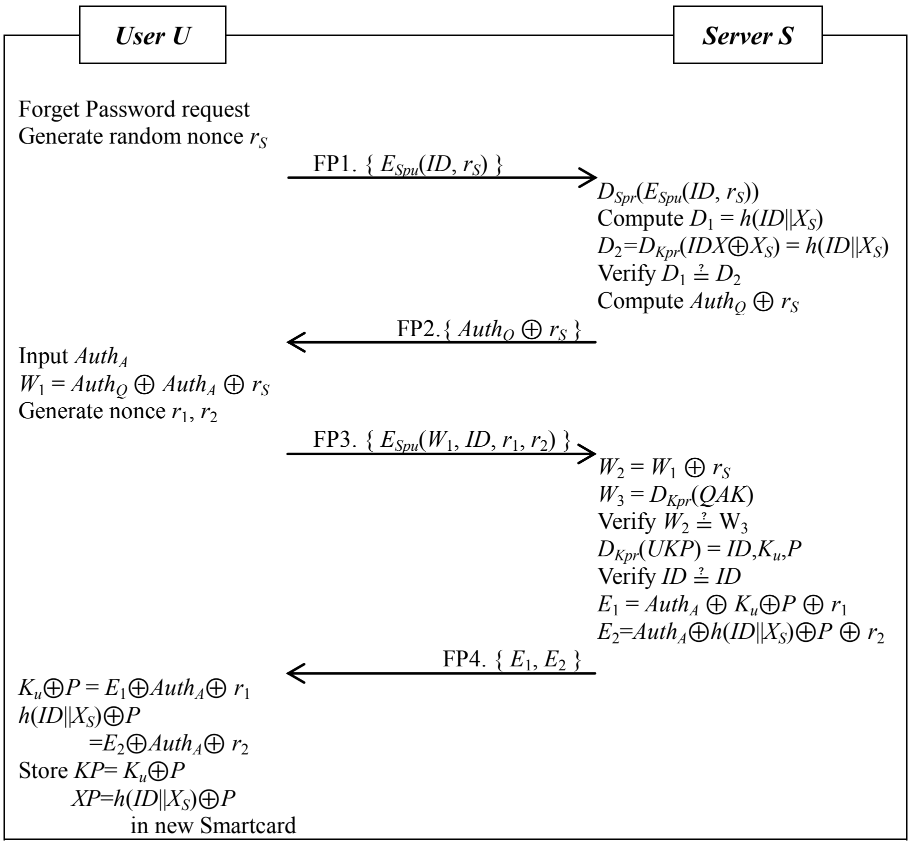


Fig. 8. Forget Password Phase

4.3 Forget Password Phase

This phase is used whenever user forgets his password. If user forget password, he/she must remember ID and $Auth_A$ at least. Fig. 8 shows the forget password phase and the detailed steps are as follows

FP1. $U \rightarrow S$: $E_{Spu}(ID, r_S)$

U inputs ID, generates a nonce r_S, encrypts them with S_{pu}, and sends the encryption result to S for a forget password request.

FP2. $S \rightarrow U$: $Auth_Q \oplus r_S$

When S receives FP1, S decrypts it with S_{pr} and computes $D_1 = h(ID||X_S)$ and $D_2 = D_{Kpr}(IDX \oplus X_S) = h(ID||X_S)$ by decrypting $IDX \oplus X_S$ with S's private key K_{pr}. S compares D_1 with D_2. If they are equal, S computes $Auth_Q \oplus r_S$ and sends it to U.

FP3. $U \rightarrow S$: $E_{Spu}(W_1, ID, r_1, r_2)$

When U receives FP2, U computes $Auth_Q$ by XORing the received FP2 with r_S. And then U inputs $Auth_A$ for the answer to the authentication question $Auth_Q$ and computes $W_1 = Auth_Q \oplus Auth_A \oplus r_S$. And U generates two nonces r_1 and r_2 and encrypts W_1, ID, r_1 and r_2 with S_{pu} and sends it to S.

FP4. $S \rightarrow U$: $E_1 = Auth_A \oplus K_u \oplus P \oplus r_1$, $E_2 = Auth_A \oplus h(X_S) \oplus P \oplus r_2$

When S receives FP3, S decrypts it and computes $W_2 = W_1 \oplus r_S$. And S decrypts QAK with S's private key K_{pr} and computes $W_3 = Auth_Q \oplus Auth_A$ and compares it with W_2. If they are equal, S decrypts UKP with S's private key K_{pr}. S computes $E_1 = Auth_A \oplus K_u \oplus P \oplus r_1$ and $E_2 = Auth_A \oplus h(X_S) \oplus P \oplus r_2$ and sends them to U. When U receives FP4, U computes $KP = K_u \oplus P = E_1 \oplus Auth_A \oplus r_1$ and $XP = h(ID||X_S) \oplus P = E_2 \oplus Auth_A \oplus r_2$ and stores them in a new smartcard. $UKP = E_{Kpu}(ID, K_u, P)$, and $QAK = E_{Kpu}(Auth_Q||Auth_A)$ in S's database. Then, S sends $h(ID||X_S)$ to U. U computes $KP = K_u \oplus P$ and $XP = h(ID||X_S) \oplus P$ and stores them in the smartcard.

4.4 Password/Verifier Change Phase

Before this phase begins, U has to perform login phase successfully. Fig. 9 shows the password/verifier change phase and the detailed steps are as follows

PC1. $U \rightarrow S$: $E_{Spu}(h(PV||K_u), r_S)$

After U finishes the Login phase, U computes $h(PV||K_u)$ using P and K_u and generates a nonce r_S. And then U encrypts $h(PV||K_u)$ and r_S with S_{pu} and sends it to S for a Password/Verifier change request.

PC2. $S \rightarrow U$: $Auth_Q \oplus r_S$

When S receives PC1, S decrypts it and computes $G_1 = h(PV||K_u)$ and $G_2 = D_{Kpr}(XPV \oplus X_S) = h(PV||K_u)$ by decrypting $XPV \oplus X_S$ with S's private key K_{pr}. And then S compares G_1 with G_2. If they are equal, S computes $Auth_Q \oplus r_S$ and sends it to U.

PC3. $U \rightarrow S$: $E_{Spu}(W_1, K_{unew}, P_{new}, ID_{new})$

When U receives PC2, U computes $Auth_Q$ by XORing the received PC2 with r_S. And then U inputs $Auth_A$ as the answer for the authentication question $Auth_Q$ and computes $W_1 = Auth_Q \oplus Auth_A \oplus r_S$. And U inputs new K_u, new P, and new ID and encrypts W_1, K_{unew}, P_{new} and ID_{new}, with S_{pu} and sends it to S. When S receives PC3, S decrypts it and computes $W_2 = W_1 \oplus r_S$. S decrypts QAK with S's private key K_{pr}, computes $W_3 = Auth_Q \oplus Auth_A$ and compares it with W_3. If they are equal, S computes $IDX = \mathrm{E}_{\mathrm{Kpu}}(h(ID_{new}||X_S)) \oplus X_S$, $XPV = E_{Kpu}(h(PV_{new}||K_u)) \oplus X_S$, and $UKP = E_{Kpu}(ID_{new}, K_{unew}, P_{new})$ using the PC3 and stores them in S's password file.

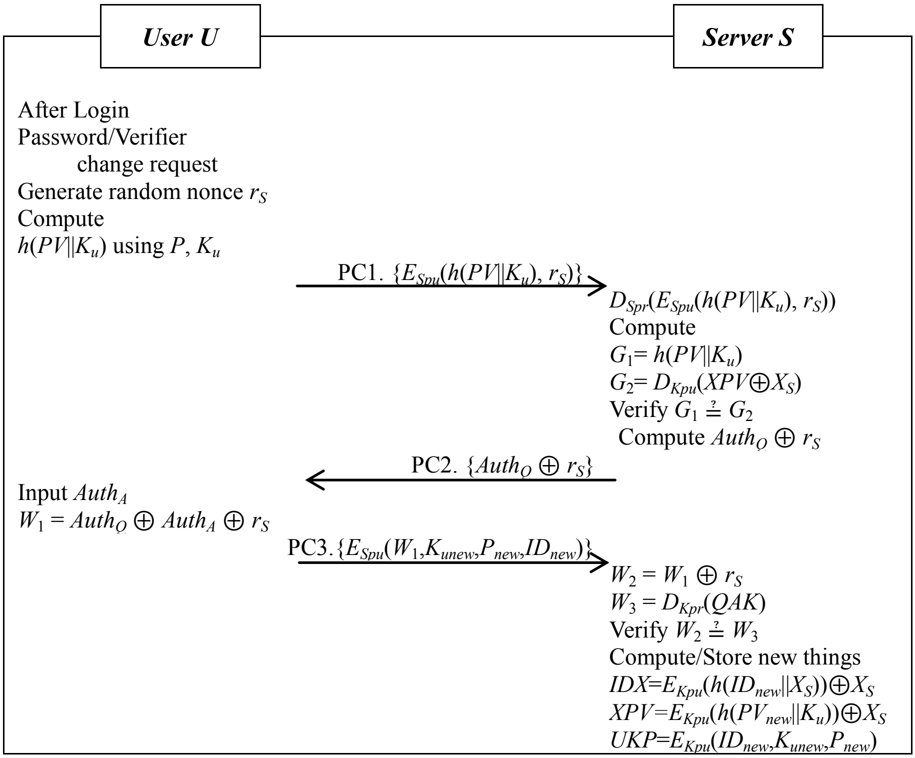


Fig. 9. Password/verifier Change Phase

5 Security Analysis

This section describes security analyses of our authentication protocol focused on password guessing attack, authentication answer guessing attack, stolen-verifier attack, impersonation attack and replay attack.

5.1 Password Guessing Attack

A guessing attack stands for an adversary's attempting to guess the user's private information. In our protocol, *U*'s private information is K_u, *P*, and $Auth_A$. *P* and $Auth_A$ differ from K_u because they are defined by *U*, which are easier than K_u against guessing attack. If the adversary has *PV* and could steal *U*'s smartcard, he/she can guess *U*'s password *P*. But the registration phase in our protocol exchanges data in secure channel and *U* does not directly send *PV* to *S* in order that the adversary does not intercept the information at login phase. So, the adversary could not guess *U*'s password in our protocol. Thus, our protocol can resist to password guessing attack.

5.2 Authentication Answer Guessing Attack

Similar with the password guessing attack, adversary can also does authentication answer guessing attack. Especially this attack is very dangerous because if adversary could successfully perform the attack, the attack could be affected to the impersonation attack as we described at Section 3. Our protocol uses random number to secure authentication question by XORing $Auth_Q$ with the nonce r_S. So, the adversary could not use the intercepted $Auth_A \oplus r_S$ for the attack because he/she does not know the nonce r_S. Thus, our protocol can resist to authentication answer guessing attack.

5.3 Stolen-Verifier Attack

For this attack, it is assumed that adversary could steal user's verifiers from server. In our protocol, the adversary can steal U's verifiers (IDX, XPV, UKP, and QAK). However, the adversary could not easily get IDX and XPV because they are secured by XORing with X_S. Specially, XPV has U's password verifier PV. Although the adversary has stolen U's smartcard, he/she could not get any useful information from the information. Furthermore, the adversary could not know $Auth_A$ by using the stolen QAK because he/she does not know $Auth_Q$. Thus, our protocol can resist to stolen-verifier attack.

5.4 Impersonation Attack

Impersonation attack stands for an adversary masquerading as a legitimate user by stealing or changing the message in a protocol. Our protocol uses challenge/response mechanism, $Auth_A$, K_u, or P to authenticate a user. If the adversary wishes to impersonate a user, he/she should know these values or send the stolen messages. However, the adversary cannot know these values because they are secured by using X_S, a hash function, or encryption with S's public keys. Moreover, if the adversary sends the stolen messages to S, S can detect that the messages were reused from the previous session by confirming the nonces in the messages. Thus, our protocol is secure against impersonation attack.

5.5 Replay Attack

The replay attack stands for an adversary storing a message in a previous session and then the adversary sending the message in the current session to masquerade as a legitimate user. Our protocol is secure against this attack. The messages in our protocol are different each session because of the usage of nonces. If S sends a nonce as a challenge to U, U should send the challenge dependent response to S. Therefore, if the adversary replays the previous message to S, S could detect the freshness of the message. Thus, the adversary cannot perform replay attack against our protocol.

6 Property Comparison

This section gives property comparisons for the security analysis and the performance analysis between related authentication protocols.

6.1 Security Comparison

Table 1 describes the security comparisons among Kim-Koc's Protocol, Jeong-Won-Kim's Protocol and our protocol. It shows that our protocol could provide securities against password guessing attack and $Auth_A$ guessing attack, stolen-verifier attack, replay attack, impersonation attack.

Table 1. Security comparisons between authentication protocols

Attacks \ Protocol	Kim et al.'s protocol	Jeong et al.'s protocol	Proposed protocol
Password Guessing attack	Insecure	Insecure	Secure
$Auth_A$ Guessing attack	Insecure	Insecure	Secure
Stolen-verifier attack	Insecure	Insecure	Secure
Replay attack	Secure	Secure	Secure
Impersonation attack	Insecure	Secure	Secure

Table 2. Computational overhead comparisons between authentication protocols

Phase \ Protocol	Kim et al.'s protocol	Jeong et al.'s protocol	Proposed protocol
Registration phase	7X+4h+2E+1D	6X+4h+4E+1D	5X+4h+4E
Login phase	11X+4h	12X+5h	11X+7h+1E+2D
Forget password phase	8X+1E+2D	12X+1E+3D	15X+2h+2E+5D
Password/Verifier change phase	11X+3h+2E+1D	8X+2h+4E+2D	9X+4h+5E+4D
X – XOR operation; h – hash function; E – Encryption; D – Decryption			

6.2 Performance Comparison

Table 2 shows computational overhead comparisons between authentication protocols. The overhead of our protocol is comparable with the other protocols. However, our protocol could achieve more security than the others. To effectively compare the protocols, registration, login, forget password, and password/Verifier change phase are considered for the analysis.

7 Conclusion

This paper has shown that Jeong-Won-Kim's authentication protocol is insecure against password guessing attack and authentication guessing attack. Then, we have proposed a secure hash-based password authentication protocol using smartcards to cope with the problems in Jeong-Won-Kim's protocol. Compared with the existing related protocols, it can be validated that our protocol is more robust authentication mechanism with better security properties than any other authentication protocols as shown in Table 1.

References

[1] Lamport, L.: Password authentication with insecure communication. Communications of the ACM 24(11), 770–772 (1981)
[2] Bellare, M., Rogaway, P.: Entity authentication and key distribution. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 232–249. Springer, Heidelberg (1994)
[3] Bellovin, S., Merritt, M.: Encrypted key exchange: password-based protocols secure against dictionary attacks. In: Proceedings of IEEE Security and Privacy 1992, pp. 72–84 (1992)
[4] Boyko, V., MacKenzie, P.D., Patel, S.: Provably secure password-authenticated key exchange using diffie-hellman. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 156–171. Springer, Heidelberg (2000)
[5] Kwon, T., Song, J.: A study on the generalized key agreement and password authentication protocol. IEICE Transactions on Communications E83-B(9), 2044–2050 (2000)
[6] Sandirigama, M., Shimizu, A., Noda, M.: Simple and secure password authentication protocol. IEICE Transactions on Communications E83-B, 1363–1365 (2000)
[7] Lin, C.L., Sun, H.M., Hwang, T.: Attacks and solutions on strong-password authentication. IEICE Transactions on Communications E84-B, 2622–2627 (2001)
[8] Chen, C.M., Ku, W.C.: Stolen-verifier attack on two new strong-password authentication protocols. IEICE Transactions on Communications E85-B, 2519–2521 (2002)
[9] Lee, C.C., Li, L.H., Hwang, M.S.: A remote user authentication scheme using hash functions. ACM Operating System Review 36, 23–29 (2002)
[10] Peyravian, M., Zunic, N.: Methods for protecting password transmissions. Computers and Security 19, 466–469 (2000)
[11] Yoon, E.J., Ryu, E.K., Yoo, K.Y.: A secure user authentication scheme using hash functions. ACM Operating System Review 38, 62–68 (2004)
[12] Ku, W.C., Chiang, M.H., Chang, S.T.: Weaknesses of Yoon-Ryu-Yoo's hash-based password authentication scheme. ACM Operating System Review 39, 85–89 (2005)
[13] Kim, Koç: A secure hash-based strong-password authentication protocol using one-time public-key cryptography. Journal of Computer and Systems Sciences International 45, 623–626 (2006)
[14] Jeong, H., Won, D., Kim, S.: Weaknesses and improvement of secure hash-based strong-password authentication protocol. Journal of Information Science and Engineering 26, 1845–1858 (2010)
[15] Wikipedia, http://en.wikipedia.org/wiki/Client%E2%80%93server_model

Towards a Better Integration of Patterns in Secure Component-Based Systems Design

Rahma Bouaziz, Brahim Hamid, and Nicolas Desnos

IRIT, University of Toulouse,
118 Route de Narbonne, 31062 Toulouse Cedex 9,
France
{rahma.bouaziz,brahim.hamid,nicolas.desnos}@irit.fr

Abstract. Security has become an important challenge in current software and system development. Most of designers are experts in software development but not experts in security. It is important to guide them to decide how and where to apply security mechanisms in the early phases of software development to reduce time and cost of development. To reach this objective, we propose to apply security expertise as security patterns at software design phase. Our methodology is based on the use of a component metamodel to capture the domain concepts and security patterns to encode solutions to security problem. The expected result is a model as design solution for specific domain. Here, we promote a modeling technique based on UML profiles to facilitate the integration of patterns solutions into model driven engineering approach (MDE). As a proof of concept, we illustrate the methodology to produce an UML profile associated with RBAC security pattern. A case study of GPS system is also provided to demonstrate the application of generated profile.

Keywords: component-based approach, security pattern, MDA, UML profile, RBAC.

1 Introduction

Software engineering has always dealt with new challenges because of both the increasing of the technology and the evolution needs of people in the current life.

Yesterday, a great challenge was to master the complexity of large systems. Component-Based Software Engineering (CBSE) allowed to reach this objective. CBSE is a discipline that is known to be reliable in the area of software engineering and that is considered to be a good solution to allow better software reuse and dynamic evolution while ensuring the quality of software.

Today, software security represents a great challenge. Indeed more and more systems are critical and require a high level of security. In such systems, failure can have serious consequences and involve direct impact on the environment or on the users. The main problem is that it becomes quasi-impossible for a standard-user (e.g. a non-security specialist) to design a secure application and even worse, it becomes very difficult for a security expert to secure an application.

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 607–621, 2011.

Thus, the main difficulty to integrate the security into a software system is: first, to know what kinds of security mechanisms to use, secondly, to know where to apply these mechanisms, and thirdly, to know what is the best level of abstraction to apply these mechanisms.

This paper proposes a methodology, dedicated to a non security specialist, for designing secure component-based application. Our methodology is based on the use of a component metamodel (that present the domain concepts) and security patterns (that present solution to security problem) to produce a specific solution to a domain (as UML profiles). We use model-driven engineering (MDE) approach to facilitate the integration of those patterns´ s solutions into the application design. To reach this objective, we base our approach on three technologies: (1) *CBSE* that aims to benefit from the reuse of prefabricated software parts called software components (2) *Patterns* as a means of exploiting hard-earned experience for a common problem (3) *MDE* for separating application domain expertise from the generic aspects. In the context of this paper we use UML profiles as an easy solution dedicated to a standard-user to apply the security at high level of abstraction.

The rest of this paper is organized as follows: Section 2 provides some reminders about tools used in our contribution. In Section 3, we introduce our approach through a short description. Then, Section 4 desribes in depth our proposition to implement security design patterns as UML profiles. As a proof of concept, we examine in Section 5 a Role Based Access Control (RBAC) pattern. In Section 6, we present the integration of the approach in a model-based development through a use case study dealing with the access control in GPS system. The paper closes with a discussion on related work in Section 7, a summary and a discussion about planned future works in Section 8.

2 Background

In this section, we will present different technologies on which we based our approach *ie.* CBSE, Patterns in software engineering and MDE for software engineering. We will justify also our choice of these methodologies.

2.1 Component Based Software Engineering

The component-based development promises to revolutionize the way applications are developed. Indeed, the introduction of this approach in the development of computer systems and especially embedded systems, represents a major step forward to reduce the complexity of systems. This approach aims at the separation of concerns by separating the business code from the management of resources. With CBSE, applications are developed with the reuse of prefabricated software parts called software components. A software component can be defined as an indivisible piece of software that cannot be deployed or used in a partial way [19]. It can be seen as a black box with required and provided interfaces. These interfaces represent services at the connection points of components. Software components are put together by connecting their interfaces.

The component approach is mainly based on the principle of reuse by assembling components that greatly simplifies the understanding and development of a system. Thus, the component reuse allows a gain in productivity because it reduces the development time of applications and enhances their strength - a component can be validated by an expert and then reused in the context of a software development process.

2.2 Patterns in Software Engineering

The concept of pattern is widely studied in literature and patterns have recently received a great deal of attention. By definition, a pattern is a proven solution to a problem in a given context [1]. There are many kinds of software patterns. These include organizational patterns, analysis patterns, design patterns [7] and object oriented patterns. We focus on the design pattern, to refer to any pattern that addresses issues to software architecture, design and implementation and more precisely we focus on *security design pattern*. Indeed, since applications have become increasingly complex and because the design of secure systems necessitates some kind of security expertise, we believe that patterns, as way of collecting and disseminating security knowledge, are a good solution to effectively convey security concepts. A design pattern can then be reused using natural languages and diagrams such as UML [13].

2.3 Model-Driven Engineering for Software Engineering

The idea promoted by MDE is to use models at different levels of abstraction for developing systems. In other words, models provide input and output at all stages of system development until the final system itself is generated. The advantage of an MDE process is that it clearly defines each step to be taken, forcing the developers to follow the defined methodology.

The proposals of the OMG around the MDE include the Model-Driven Architecture (MDA) framework [4] [12] which allows an implementation of MDE through a set of standards.

In MDA, everything is considered as a model or a model element; hence, it is important to use a well-defined modeling language, such as UML, to describe each model precisely. However, the existing standard modeling languages are so general as to limit their suitability to model a specific domain [6]. To resolve this issue, the MDA enables the user to define domain-specific languages by providing two alternative approaches. The first approach is to use the Meta-Object Facility (MOF) [11] to define a new modeling language. The second approach is to extend the UML language with domain specific information, as UML profile.

2.4 Role in Secure Component-Based Systems Design

In this paper, we will use the CBSE paradigm because we assume that design by assembling components is a promising approach in the domain of embedded systems and a promoted way to manage complexity of these systems. CBSE enables these systems to be easily assembled and less costly to be build. In the object of secure component applications at high level of system design, we will use security expertise as design patterns called security patterns (used to specify security requirements). We believe that patterns, as way of collecting and disseminating security knowledge, are a good solution to effectively convey security concepts. These two technologies have many common and complementary aspects. That's why, we assume that the combination of design patterns and the component approach further improves reuse, quality and facilitate the way in which security is added at high level of software development. To facilitate the integration of those patterns´ solution into the application design and implementation, we use the MDA approach. We use the MDA process to represent the security aspect in all stages of a system development. We propose an application of security patterns at high level of abstraction based on the semantic defined by the UML profile. Using the UML profile to model a specific domain will avoid us many problems as (1) the definition of a new set of semantics and notations to address the domain specific characteristics (2) these new semantics will not be compatible with commercial UML tools (3) users need to learn these new notations and semantics to master this new language. All these disadvantages, justify our choice of the approach that extend the UML language with an UML profile.

3 Our Approach in a Nutshell

We assume that security has become an important challenge in current software and system development also we assume that most of designers are experts in software development but not experts in security. It is important to guide them to decide how and where to apply security mechanisms in the early phases of software development to reduce time and cost of development.

Hence, we propose, in this paper, a methodology dedicated to a non-security specialist, for designing secure component-based application. Our methodology is concretely made by combining a component meta-model with a security pattern to produce a security UML profile that represents a specific solution to a domain. Thus, our approach implies that the model-driven engineering approach allows the flexible integration of security patterns' solution that embodied security expertise into the application design and implementation. Therefore, the application designers/developers can add security to their applications in an easy and natural way. The better integration of design patterns in component based systems design is based on applying semantics into models provided by the UML profiles. So in this way, when security patterns are chosen (according to the security requirements for the particular application) at the abstract level, then the corresponding concrete security profile can be generated. In some way, we encourage developers to integrate security expertise to their applications metamodels by choosing appropriate or desired variant of security patterns. In this phase developers do not have to bother themselves about the concrete details of patterns. Consequently, the application of the solution proposed by the desired pattern consists only of applying security profile.

We propose the framework represented in Fig. 1 to illustrate our approach that aims to decouple the application domain expertise from the security expertise and solutions that are both needed to build a secure application.

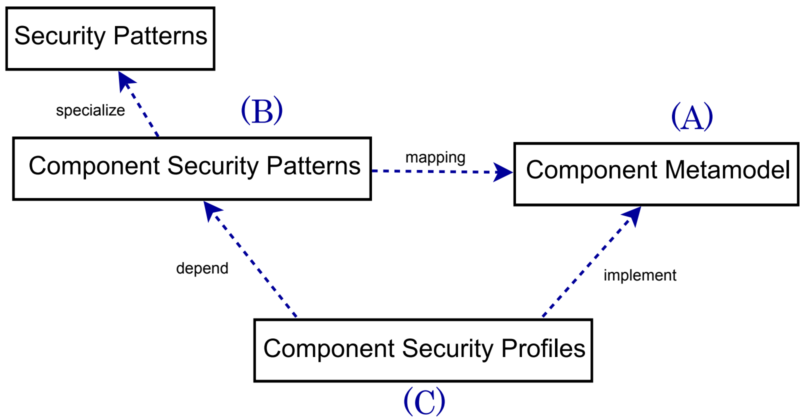


Fig. 1. Overview of the framework

The proposed framework is composed of the following parts:

- *Nodes*
  - (A) *Component Metamodel* defines generic concepts of component approach;
  - (B) *Component Security Patterns* contains a specification of *Security Patterns* adapted and adjusted to the component concepts presented in the *Component Metamodel*;
  - (C) *Component Security Profiles* implements generic component concepts detailed in the *Component Metamodel* and uses solutions provided in *Component Security Patterns*.
- *Edges*
  - *specialize*: adapt security pattern to the component based context;
  - *mapping*: this link presents the mapping between the Component Metamodel and the Component Security Pattern to produce Component Security Profiles;
  - *implement*: UML Component Security Profile is an implementation of the Component Metamodel;
  - *depend*: UML Component Security Profile implements the Component Metamodel depending on the solution provided by the Component Security Pattern.

Each level is discussed in the next subsections. We sketch a few notable example next. For clarity's sake, we use UML notations to describe such levels.

4 Methodology for Producing the Security Profile

The goal is to construct a common way to implement secure applications using security patterns at design level for several domains. Now we explain in depth the three layers.

4.1 Component Metamodel

The component metamodel, presented in Fig. 2, is generic in the sense that it describes the fundamental concepts that are already used with success in CBSE (Component-Based Software Engineering) [19]. This metamodel is the basic building block of our process. It will be extended in the next section to support security aspect proposed by the security patterns. In what follows, we detail the concepts of such a metamodel:

- *Component* is one resolution unit with an identifier, a name, a type and a value. It can provide and need a set of services. In other words, it can be seen as a black box that provides and requires services through *PointOfConnection*.
- *PointOfConnection* is attached to a *component* to interact with its environment. A *component* defines one or more component. A *PointOfConnection* has the following properties : an identifier, a name and a type.
- *ProvidedPointOfConnection* is a *PointOfConnection* that specifies which properties are provided by the component to be used by other component of its environment.
- *RequiredPointOfConnection* is a *PointOfConnection* that specifies which properties the component must found in its environment to work properly.
- *Connection* it is a link that connects a *ProvidedPointOfConnection* to a *RequiredPointOfConnection*. A component assembly defines a set of connections.
- *NFP* is a concept to capture *Non-Functional Properties* related to communications.

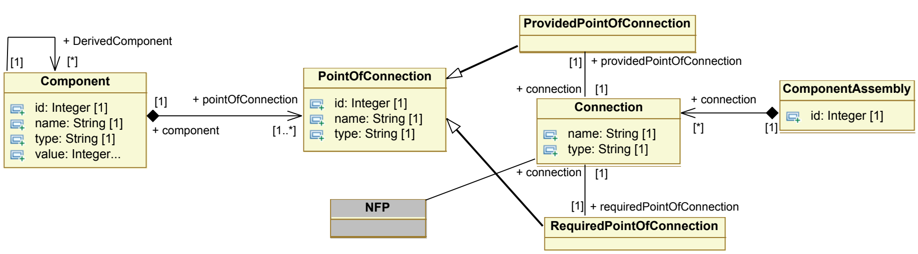


Fig. 2. Component Metamodel

4.2 Component Security Pattern

The component security pattern is a chosen security pattern according to the security requirements for the particular application. Recently there has been an increasing interest about developing those type of security patterns and in a short period of time, many proposition of security patterns appeared.

Table 1. A subset of security patterns

Security Pattern	Intent
1. Authenticator [16]	Identity verifier of the subject.
2. Authorization [16]	Access policy definition for resources.
3. Defense in Depth [20]	Add security checks in multiple layers of the application.
4. Full Access with Errors [16]	Users can see all the options but can only access the options they are authorized to perform.
6. Policy Enforcement Point [16]	Concentrates Identification and Authorization mechanisms at a point.
7. Replicated System [2]	Components are replicated to distribute workload.
8. Role Based Access Control (RBAC) [15]	Control access to resources based only on the role of the subject.

Table 1 lists samples of these patterns along with information about their source and a short description of their intent. These patterns are majority presented in the object context. In our work, we will investigate the adaptation of those security patterns to the component context.

The description template used in this part considers six fundamental elements including: *name*, *intent*, *context*, *problem*, *structure* and *participants*. The *name* of the security pattern facilitates the discussion and documentation. Its *intent* describes succinctly the problem being the pattern applies and useful. The *context* the situation in which the pattern may apply. The *problem* describes a problem that needs an appropriate solution and what conditions have to be met in order to be able to use the pattern. The *structure* indicate with class, sequence, and other UML diagrams, the form of the solution. Finally, the *participants* describe participants in the solution proposed by the patterns.

To illustrate our approach, the next section deals with the RBAC pattern to secure access to services provided by software component.

4.3 Component Security UML Profile

The component security profile represents a specific solution to a domain. It embedded security expertise provided by the security pattern presented in the previous subsection. This profile is resulting from the component metamodel that present the domain concepts (component approach basic concepts) and the choosing security pattern that present security solution according to the problem faced.

Fig. 3 depicts the construction process of the proposal. That is based on three parts : (1) *Component Metamodel* that present the domain concepts, (2) *Component Security Patterns* that present solution to security problem. The solution produced from these two first parts is (3) *Component Security Profile* that represent a specific solution to a domain.

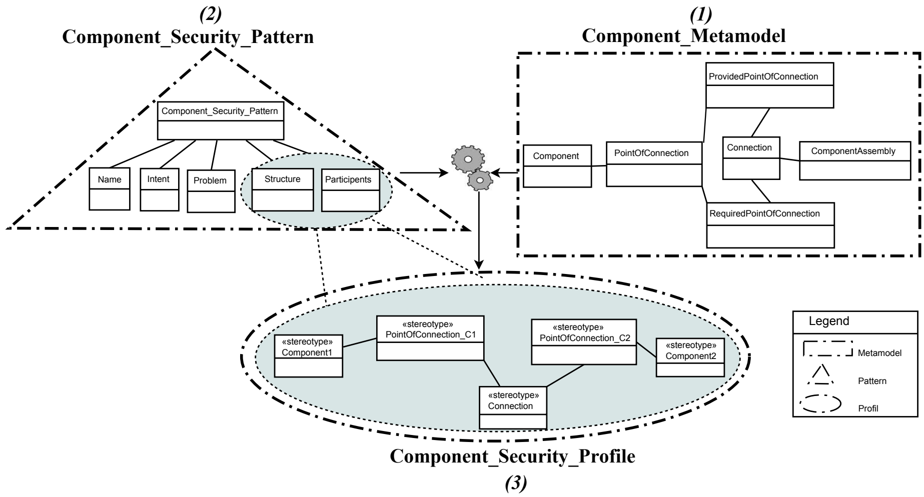


Fig. 3. Construction process of the proposal

Now, we will present an overview of our algorithm. Then, we describe the principals steps of the algorithm in more detail through an example using RBAC pattern. The algorithm is composed of the following steps:

Algorithm 1. Profile Construction Process

- **Input :** A generic component meta model and a chosen security pattern.
- **Results:** A UML profile
- **Rules:**
  - Extends the component Meta Model using UML notions to capture the related security concern
  - Identify the potential place to apply a Pattern
  - Identify the mapping between the roles of the patterns and the component platform independent model
  - Produce an UML Profile
  - Configure the Pattern for a Component Paltform Specific Model

5 Illustration of Our Proposal Using RBAC Pattern

To illustrate our approach, we use RBAC pattern to solve the problem of access control to services provided by software components. In the following subsections we will present *Component-Based Metamodel Extension* to support access control aspect, a *Component RBAC Pattern* as an example of security patterns and the *UML profile* generated from the metamodel and the chosen security pattern.

5.1 Component-Based Metamodel Extension

To support access control aspect the component-based metamodel has been extended. In this paper, we choose to manage security at connection level because we assume that this strategy allows to integrate security without modify the initial component model. This extension is presented in Fig. 4.

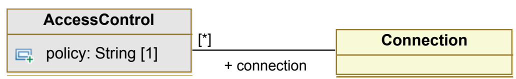


Fig. 4. Component-Based Metamodel Extension

- *AccessControl* artefact corresponds to the protection of the component in the invocations of other components that we specialize in component point of connection. The access control is implemented by the policy approaches.

5.2 Component RBAC Pattern

We use the Component RBAC Pattern [15] presented in Fig. 5, we can see that the RBAC pattern has not exactly the same structure as proposed in [15] this is because we adapt it to the component-based context. This pattern offers a solution adopted to solve the problem of access control to services provided by software components.

Intent. Control access to services provided by software components based only on the role of the *CallerComponent*.

Context. Any environment where we need to control access to services provided by software components and where components can be classified according to their tasks.

Problem. Permissions for *CallerComponent* accessing *ProtectedComponent* have to be described in a suitable way.

Structure. Fig. 5 illustrates the structure of this security pattern using a UML class diagram.

Participants. The class *Component* describes a generic component. The classes *CallerComponent* and *ProtectedComponent* inherit from the *Component* class. The classes *CallerComponent* and *Role* describe respectively, the component requesting a service and pre-defined roles. The association class defines the *Right* types of access rights for the component *ProtectedComponent*.

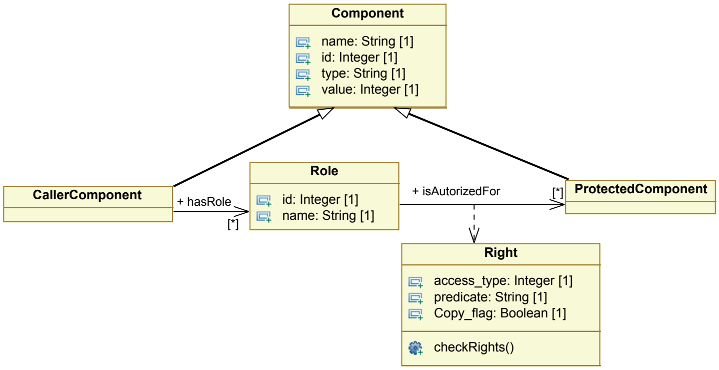


Fig. 5. Component Role-Based Access Control Pattern

5.3 Component Access Control UML Profile

We propose an implementation of *Component Access Control Profile* from the *Component-Based Metamodel* and the *RBAC pattern* described in the two previous sections. This UML profile can specify access control to software components. We choose to apply solution proposed by the RBAC pattern at *PointOfInteraction* level. So both *RequiredPointofConnection* and *ProvidedPointofConnection* stereotypes extend the UML metaclasse *Interface* and the operation of checking rights is at the level of the stereotype *ConnectionWithAccessControl*.

The design of *Component Access Control Profile* definition is shown in Fig. 6. It is composed by the following stereotypes:

– The stereotypes *CallerComponent* and *ProtectedComponent* substitutes the metaclasse *Component* in our metamodel and extend the UML *Component* metaclasse.

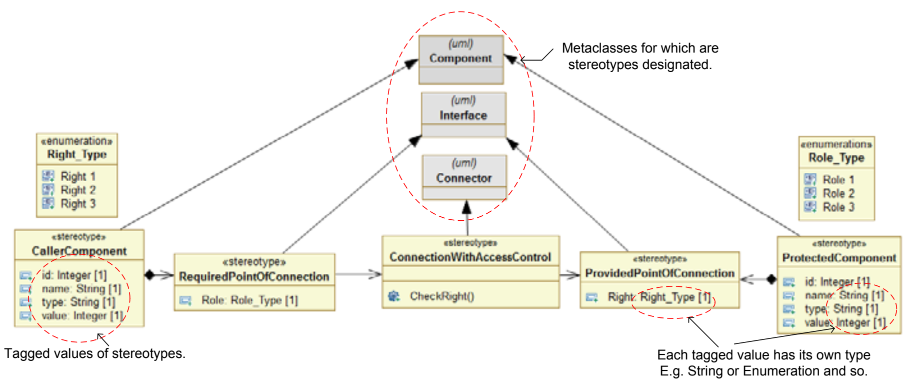


Fig. 6. Component Access Control Profile

- The stereotype *ConnectionWithAccessControl* substitutes the metaclass *Connection*.
- Both *RequiredPointofConnection* and *ProvidedPointofConnection* stereotypes are respectively annotated by the properties *Role* and *Right*.

6 Integration on a Model Based Development Approach

In this section, we propose a guideline based on an MDE approach to build secure component applications through a notable example with Access Control requirements: a GPS. For simplicity's sake, many functions of this use case have been omitted.

6.1 Description of the Basic GPS System

The Basic GPS consists of three parts: The *space segment*, The *control segment* and The *user segment*. In what follows, we are interested in the space segment and the user segment. Indeed, in these two parts we have identified requirements for access control to services offered by components. In this example, we consider mainly the management of access control to various services offered by phone operators especially downloading geographic maps in real-time and manage secure access to satellites. The Basic GPS system described above works as follows:

(1) The *GPS Terminal* receives continuously the signal of *Satellite* as well as that of *SecureSatellite*. The *SecureSatellite* is active if the user has access rights to it.
(2) The *GPS Terminal* sends a request to download map to the *Phone Operator*.
(3) The *Phone Operator* allows the user, depending on it's access rights, to download the requested geographic map.

6.2 The Components System Basic GPS

With regard to the Basic GPS system, we have identified the following use cases:

- Downloading geographic maps
- Access to secure satellites.

To implement these two use cases, we propose to use components described by the UML component diagram as shown in Fig. 7. These components are:

- *Satellite* enables to emit permanently a navigation message containing all the necessary data for the receiver to perform all the navigation calculations. This service is provided through the interface *ISatellite*.
- *SecureSatellite* emits secure signals via the interface *ISecureSatellite*.
- *GPSTerminal* receives the message transmitted by the *Satellite* through the interface *ISatellite* and must have access rights to the signals sent by *SecureSatellite*. It requires the map downloading service from the *Operator* component through its interface *IOperator*.

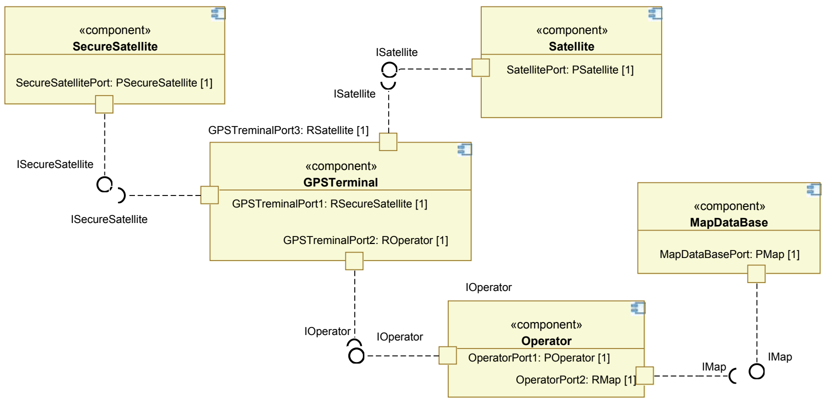


Fig. 7. The components system Basic GPS

- *Operator* (*Phone operator* abbreviation) offers the service to download maps through the interface *IOperator*. It requires maps requested by members via the *IMap* interface from *MapDataBase*.
- *MapDataBase* offers the possibility to the *Operator* to have the map download by the applicant via the interface *IMap*.

6.3 Basic GPS System Architecture

In this section, we apply the formalism of composite structure UML diagram to model the system Basic GPS. To describe the access control to *SecurSat* component (instance of *SecureSatellite*), *Server* component (instance of *Operator*) and *DB* component (instance of *MapDataBase*), we use the *Component Access Control Profile* defined in Section. 5.3. These components and their ports (see Fig. 8) are respectively annotated by *ProtectedComponent*, *CallerComponent* and *ConnectionWithAcessControl* stereotypes.

7 Related Work

Many studies have already been done on modeling security in UML. The work presented in [8] introduces an extension UMLsec of UML that enables to express security relevant information within the diagrams in a system specification. UMLsec is defined in form of a UML profile using the UML standard extension mechanisms. Another approach in [9] provides a modeling language for the model-driven development of secure, distributed systems based on UML. Their approach is based on role-based access control with additional support for specifying authorization constraints. SecureUML is a modeling language that defines a vocabulary for annotating UML-based models with information relevant to access control. These two approaches are not in competition but they complement each other by providing different viewpoints to the secure information system.

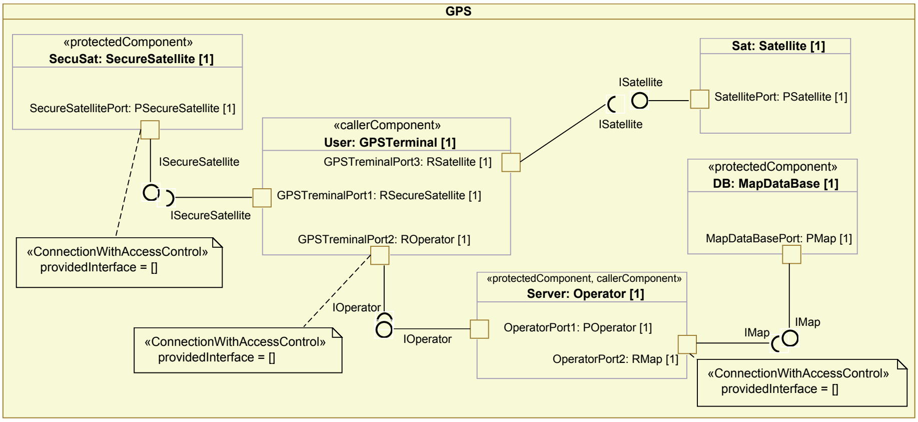


Fig. 8. Architecture-based system components Basic GPS

Regarding the modeling of security, a set of patterns of security has been proposed as applicable to the process of developing applications. In that directive, [17] proposed in his book different patterns for managing access rights in a computer system including: a system of patterns for managing access control to a system and a template for control access to database roles, patterns of access control based on metadata. Other works, like those proposed in [15] were particularly interested in patterns of security for managing access control. The work of [22] includes a catalogue of models in the form of design patterns that can be applied during the introduction of security in application development. An overview of a large family of patterns for security is presented in [23]. An application of these patterns in different stages of development is also available with an illustration of their integration.

In addition to the work outlined in this section, some works of literature consider the combined treatment of security and other non-functional properties. However, these studies do not offer new ways to take these issues into account during the design phase of applications and systems.

Concerning patterns instantiation and application, several proposals have already been made. The works presented in [18] presents an approach that uses a formal design pattern representation and a design pattern instantiation technique of automatic generation of component wrapper from design pattern. Additionally, there exists several approaches introducing their own tool based support for the pattern instantiation. In [3] the authors proposes an approach for representing and applying design patterns. [21] provide a simple UML profile for design pattern which allows the explicit representation of design patterns in UML models with a workable model transformation approach and a transformation process. Another method was introduced by [5]. They have developed a methodology for the creation of automated transformations that can apply a design pattern to an existing program. In [14] the authors propose a method to design pattern application support in software project. The elaborated method is based on semantics defined via UML profile and transformation of models.

In contrast to the above approaches, our method addresses the non-functional requirements separately from the functional requirements by embodying the security knowledge from safety experts, in the form of safety patterns. Our contribution includes to adapt proposed security patterns, that capture security knowledge, to the component based context then integrate these patterns to UML models as UML profiles. We use UML profile as an extension of UML metamodel to benefit from its compatibility with majority of other UML tools.

8 Summary and Future Work

In this paper, we present a methodology that consists of combining a component metamodel with a security pattern to produce a security UML Profile that represents a solution specific to a domain usable by a standard-user. The presented approach focuses on application of design patterns at a higher level of abstraction by combining a generic component-based metamodel with a security pattern in order to secure systems. In other words, we address the non-functional requirements separately from the functional requirements by embodying the security knowledge from security experts, in the form of security patterns. We are proposing a novel approach to secure component-based applications based on security patterns and component-based approach to further improves software quality and software development productivity. Since security must not be considered as an isolated aspect, but as an aspect present in all stages of a system development, we rely on the MDA approach to integrate security. Our approach is based on standard extension of UML metamodel via an UML profile.

As extensions of our work, we will investigate other security patterns to build new UML profiles and take into account other non-functional properties such as dependability. We still need to refine this approach, in particular, we want to formalize rules needed to integrate patterns in high level of abstraction and choose the best classification needed to organize patterns in security patterns repository. Such an organization must show the users the distinct characteristics of the different patterns to support them to detect the suitable pattern needed for the application context. Further research needs to be done on selecting design pattern based on user requirement and security level needed and application specific constraint. We plan to develop a tool supporting the overall approach proposed in this paper and relying existing tools and related standards such as MARTE profile [10].

References

1. Alexander, C.: A Pattern Language: Towns, Buildings, Construction.. Oxford University Press, USA (1977)
2. Blakley, B., Heath, C.: Security design patterns technical guide - version 1 (2004)

3. El Boussaidi, G., Mili, H.: A model-driven framework for representing and applying design patterns. Annual International Computer Software and Applications Conference 1, 97–100 (2007)
4. Bézivin, J.: Towards a precise definition of omg/mda framework. In: Proceedings of ASE 2001, pp. 273–280. IEEE Computer Society, Los Alamitos (2001)
5. Cinnéide, M.Ó., Nixon, P.: Automated software evolution towards design patterns. In: Proceedings of the 4th International Workshop on Principles of Software Evolution, IWPSE 2001, pp. 162–165. ACM, New York (2001)
6. Fuentes-Fernández, L., Vallecillo-Moreno, A.: An Introduction to UML Profiles. UPGRADE, European Journal for the Informatics Professional 5(2), 5–13 (2004)
7. Helm, R., Ralph, E., Johnson, R., Vlissides, J., Gamma, E.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional Computing Series (1995)
8. Jürjens, J.: Umlsec: Extending uml for secure systems development. In: Jézéquel, J.-M., Hussmann, H., Cook, S. (eds.) UML 2002. LNCS, vol. 2460, pp. 412–425. Springer, Heidelberg (2002)
9. Lodderstedt, T., Basin, D., Doser, J.: SecureUML: A UML-Based Modeling Language for Model-Driven Security (2002)
10. OMG. Uml profile for marte. In OMG document ptc/07-08-04 (2007)
11. OMG. Omg metaobject facility (mof) specification, version 2 (2010)
12. OMG. Omg model driven architecture specification: Mda guide version 1.0.1 (2010)
13. OMG. Omg unified modeling language specification version 2.3 (May 2010)
14. Kajsa, P., Majtás, L.: Design patterns instantiation based on semantics and model transformations. In: van Leeuwen, J., Muscholl, A., Peleg, D., Pokorný, J., Rumpe, B. (eds.) SOFSEM 2010. LNCS, vol. 5901, pp. 540–551. Springer, Heidelberg (2010)
15. Priebe, T., Fernández, B., Ingo Mehlau, J., Pernul, G.: A pattern system for access control, pp. 235–249. Kluwer, Dordrecht (2004)
16. Schmidt, C.: Guest editor's introduction: Model-driven engineering. Computer 39, 25–31 (2006)
17. Schumacher, M.: Security Engineering with Patterns - Origins, Theoretical Models, and New Applications. LNCS, vol. 2754. Springer, Heidelberg (2003)
18. Stephen, S., Dong, N.: Integration in component-based software development using design patterns. In: Annual International of Computer Software and Applications Conference, p. 369 (2000)
19. Szyperski, C., Gruntz, D., Murer, S.: Component Software Beyond Object-Oriented Programming. Addison-Wesley/ACM Press (2002)
20. Viega, J., Mcgraw, G.: Building Secure Software: How to Avoid Security Problems the Right Way. Addison-Wesley Professional Computing Series. Addison-Wesley Professional, Reading (2001)
21. Wang, X., Yuan Wu, Q., Min Wang, H., Xi Shi, D.: Research and implementation of design pattern-oriented model transformation. In: International Multi-Conference on Computing in the Global Information Technology, p. 24 (2007)
22. Yoder, J., Barcalow, J.: Architectural patterns for enabling application security. In: PLoP (1997)
23. Yoshioka, N., Washizaki, H., Maruyama, K.: A survey of security patterns. Progress in Informatics (5), 35–47 (2008)

Enriching Dynamically Detected Invariants in the Case of Arrays

Mohammadhani Fouladgar, Behrouz Minaei-Bidgoli, and Hamid Parvin

School of Computer Engineering, Iran University of Science and Technology (IUST), Tehran, Iran
Hani_fouladgar@comp.iust.ac.ir, {b_minaei,parvin}@iust.ac.ir

Abstract. Software engineering is the work of designing, implementing and modifying of software to build software fast and have a high quality, efficient and maintainable software. Invariant helps programmer and tester to perform some aspect of software engineering. Since arrays and pointers are more probable to be faulty, invariants which report these properties are more useful. By presenting first and last elements of arrays we can detect the carelessness in using index which mostly happens in loops. The idea of employing number of mutual elements between two same type arrays can help us to catch faults in the cases that an array is gained from changes in another array. These two properties help us to capture lots of prevalent faults and therefore try to remove them. This paper proposes an interesting sort of extension over Daikon like tools. It can generate more invariant in the case of array.

Keywords: dynamic invariant detection; software testing; array property; array's first and last elements; mutual element between arrays.

1 Introduction

Invariants are some properties of different program points which are true in all executions of the program. These properties can be seen in *formal specification* or *assert statement*. For example in a sort program that its job is to sort array of integers, invariant (array **a** sorted >=) is reported. This idea is introduced in [1]. Invariants develop data structures and algorithms and are employed in all aspects of software engineering from design to maintenance [2]. Invariants represent program properties; hence after an update to the code, invariants can show properties which remain unchanged or those that are violated through code revision. Invariants, somehow, are treated such as *documentation* and *specification* of a program. Specification is used in all software engineering steps like design, coding, verification, testing, optimization and maintenance. Invariants might be detected by *static* and *dynamic* approaches.

Static analysis checks syntactic structure and runtime behavior of program without actually running [3]. Static analysis is an entirely automated method. *Data-flow analysis* is a traditional technique which is employed in the compilers in order to collect necessary information for optimization. Data-flow analysis can determine the properties of program points. In practice the collected information is considered

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 622–632, 2011.

invariant. *Abstract interpretation* is a theoretical framework for static analysis [4]. The most precise imaginable abstract interpretation is called the *static semantics* or *accumulating semantics*. Another related static analysis technique is *predicate abstraction*. In this technique, an absolute and finite set of predicates is used to induce abstract domains.

Dynamic invariant extraction from program context is brought to software engineering realm during recent ten years. In contrast to static analysis, dynamic invariant detection extracts program properties and information by executing of the program with different inputs. Dynamic invariant detection, first time, was quoted by *Daikon* [2] - a full-featured and robust implementation of dynamic invariant detection. With the help of test suits and different executions of a program as well as using an invariant inference system, properties of specific point of program (usually function entries and exits) are revealed. These points, lexically, are named *program points*. The properties are not certainly true, but indeed, are determined through several executions on test cases with specific confidence. An important attitude of process of dynamic invariant extraction is that what invariant presents is not only program behavior but also indicates assumption of test cases. This nature of invariant causes double usage of it, so dynamically-produced invariant determines program specification more properly.

This paper concentrates on the dynamic extraction of invariants. We introduce some ideas which improve the effect of invariants that could be useful in bug detection and testing. We propose two extensions to the Daikon like tool for invariant detection. The extensions are concerned with the treatment of arrays. We employ more properties of arrays to get better results. The rest of this paper starts with related work (section 2) and continues with contributions (section 3). Then we bring some simple examples to clarify the ideas (section 4). In section 5 some actual examples and adjustment for our ideas are presented. We evaluate our idea in section 6. Finally we conclude the paper and talk about future work (section 7).

2 Related Work

In this section, we discuss some implementations of dynamic invariant detection. Many valuable efforts have been done in this field but we mention here only these ones which are more relevant.

Dynamic invariant detection, as mentioned, is quoted by *Daikon* [2]. Daikon is the most prosperous software in dynamic invariant detection developed until now, comparing with other dynamic invariant detection methods [2]. However this software has some problems out of which the most serious one is being time-consuming.

DySy proposes a dynamic symbolic execution technique to improve the quality of inferred invariant [7]. It executes test cases like other dynamic invariant inference tools but, as well, coincidentally performs a symbolic execution. For each test unit, DySy results in program's path conditions. At the end, all path conditions are combined and build the result.

Software Agitator is a commercial testing tool which is represented by Agitar and is inspired by Daikon [5]. Software agitation is a testing technique that joins the results of research in test-input generation and dynamic invariant detection. The results are called *observations*. Agitar won the Wall street Journal's 2005 Software Technology Innovation Award.

The DIDUCE tool [6] helps programmer by detecting errors and determining the root causes. Besides detecting dynamic invariant, DIDUCE checks program behavior against extracted invariants up to each program points and reports all detected violations. DIDUCE checks simple invariants and does not need up-front instrument.

3 Paper Contributions

Software testing is one of the most time consuming parts of software engineering because regarding inputs, different executing paths happen and unchecked paths can be defective. In this situation, because of their structure, *arrays* and *pointers* are more probable to be faulty. In the C, an array is a kind of pointer. Therefore if any improvement is achieved for the handling of arrays, it can be simultaneously considered as an improvement for the pointers.

The first and the last elements of an array possess very crucial properties because these elements are impacted by the carelessness in using the indexes. By involving some array elements in invariant detection, a dramatic improvement in fault detection might happen. The number of these elements can be the least size of an array or they can be optional. This contribution exposes inattention in using index which mostly happens with the first and the last indexes and corresponding to the first and last elements of an array.

Besides employing array elements, enlisting the number of mutual elements of same type arrays for each program point is useful in detecting faults. In other words, for each program point, the number of elements' values which are shared in two different same type arrays is employed in invariants detection. It helps the programmer to evaluate his program in the cases that an array is gained from changes in another array. The mutual elements show the correct elements which should be unchanged through the process. We discuss more about this contribution in the next sections and clarify the number of mutual elements of same type arrays for each program point.

Overall our contributions comprise the following:

- Adding the number of first and last elements of an array as new variables for invariant detection.
- Enlisting the number of mutual elements of same type arrays for each program point

4 Illustration of Contributions

To clarify the contributions we discussed earlier, we present some pieces of program code and their post-condition invariant. The programs are in the form of pseudo-code and do not assume the use of any specific programming. For presenting our ideas, the *Exit* program point invariants which represent post-condition properties for a program point are used because post-condition properties can show both the pre-condition and post-condition values of variables.

4.1 First and Last Elements of Array

For invariant detection, we might propose the use of the use of first and last elements of an array. The number of these elements can be the least size of the array or can be optional. This contribution exposes carelessness in using index which mostly happens to first and last indexes.

Fig.1 presents the code of bubbleSort() function. It accepts 2 as input one of which is the array and another is the length of the array. The output is the sorted array. In this example, the index *j* starts at 1 instead of 0 so the first element of array is not considered in the sorting. With the contribution of the produced invariants from the first and last elements of the array the fault is detected. Adding the first and last elements of each array, as part of the related invariants in the *Exit* point of the bubbleSort() function is illustrated in Fig.2.

```
int *bubbleSort(int *digits,int length)
{
  int *numbers;
  number <- digits;
  for(i=1;i<length;i++)
    for(j=1;j<length-i;j++)
      if(numbers[j]>numbers[j+1])
      {
        int temp=numbers[j];
        nembers[j]=numbers[j+1];
        numbers[j+1]=temp;
      }
  return numbers;
}
```

Fig. 1. Program A: Inattention in using index

The presented invariants in Fig.2 are in the form of Daikon output. For array *x*, *x[-1]* is the last element of *x*, *x[-2]* the element before the last one and so forth. In Fig.2, line 14 shows that the first element of the input array always equals to the first element of the return value. Lines 15 to 20 show that the rest of the elements are sorted. Therefore obviously only the first element is never involved in sorting. This helps the programmer to detect the fault.

4.2 Number of Mutual Elements between two Arrays

Another property, which can help the programmer, might be the number of mutual elements of same type arrays for each program point. It is helpful for a programmer to test the code in situations that an array comes from another one. To illustrate the idea you may consider Fig.3.

Fig.3 is a function which accepts 4 parameters as input. The first parameter is a sorted array and others are respectively array length, the value of element which must be replaced, and the new value, respectively. This function replaces *m*'s value with *n*'s value as a new value.

```
digits[] >= return[] (lexically)
digits[] == orig(digits[])
orig(length) == size(return[])
return != null
return[] elements <= return[-1]
digits[] elements <= return[-1]
return[1] in digits[]
return[2] in digits[]
return[3] in digits[]
return[-1] in digits[]
return[-2] in digits[]
return[-3] in digits[]
return[-4] in digits[]
return[0] == digits[0]
return[1] < return[2]
return[2] < return[3]
return[3] < return[-4]
return[-4] < return[-3]
return[-3] < return[-2]
return[-2] < return[-1]
length != return[0]
```

Fig. 2. Related invariants to the code of Fig 1

Exit point invariants of replace() is shown in the Fig.4.

```
void replace(int *d,int l,int m,int n)
{
  int i;
  for(i=0;i<l;i++)
   if(d[i]==m)
   {
     d[i]=n;
     break;
   }
}
```

Fig. 3. Program B: An example of "replace code"

In Fig.4 invariants in lines 6 and 7 show the number of mutual elements between *d* (the first parameter of the function) and the return value. As expected, the number of mutual elements between *d* and the return value equals to the number of mutual elements between *orig(d)* and the return value. Also, the number of mutual elements between *d* and the return value equals to the size of *d* minus 1. The latter confirms the validity of the program since the number of mutual elements equals to the size of *d* minus 1. However, besides this invariant, other invariants quote that the return value is not sorted despite *d* is sorted and this might be a fault in the program.

```
d[] == orig(d[])
orig(l) == size(return[])
d[] sorted by <
return != null
orig(m) in d[]
Mutual(d[] , return[]) == Mutual(orig(d[]), return[])
Mutual(d[] , return[]) == size(d[])-1
d[] elements <= d[-1]
orig(n) in return[]
orig(l) < d[-1]
orig(l) < return[-1]
orig(m) != size(d[])-1
orig(m) < d[-1]
orig(m) != return[-1]
orig(n) != d[-1]
d[-1] >= return[-1]
```

Fig. 4. Related invariants to the replace code of Fig 3

5 Actual Examples and Justification

In this section we describe our ideas by using some programs as actual examples. These examples show how our ideas help the programmer to detect the faults and subsequently properly address them. These examples comprise small and rather simple subprograms. Our reasonable assumption is that every program, either big or small, can be divided in small parts and might be raised in small subprograms. In other words, in all programs, when working with arrays the programmer uses iteration expressions such as the "*for"* block and carelessness can result independently of whether the program is big or small. Therefore to illustrate our justification we bring some actual but not large subprograms, which are caused "gold standard" invariants [9]. The goal of adding these examples is to validate our ideas and to display the feebleness of the previous efforts. Like in the previous section, the presented code does not assume the use of any specific programming

5.1 Try-Catch and Effectiveness of the Ideas

We start with function AVG() which contains a *Try-Catch* statement. This function is presented in Fig.5. It accepts an array (a[]) and the length (l) and sums all the elements in sum, then divides each array element by n/5 and finally returns the sorted array. Although the programmer has considered that if n is zero a division-by-zero happens and prevented it from happening by introducing an if-condition statement, the code has a fault. "temp" has been declared as an integer and for $0<n<5$, n/5 is zero subsequently the variable temp can become 0 as well, and therefore division-by-zero

```
int* AVG(int* a,int l,long* sum,int n)
{
  int i,j,*numbers,temp;
  numbers=malloc(sizeof(int[l]));
  numbers[]<-0;
  Try
  {
    *sum=0;
    for(i=0;i<l;i++)
    {
      *sum=a[i]+*sum;
      if(n!=0)
      {
         temp=n/5;
         a[i]/=temp;
      }
    }
    numbers=Sort(a,l);
  }
  Catch(e)
  {
    *sum=0;
  }
  return numbers;
}
```

Fig. 5. Program C: First example for the justification of the proposed algorithm

happens. In these situations a division-by-zero exception is thrown and the return array has all its elements equal to 0 instead of being the sorted input array.

In Fig.6, we present the related invariants in the *Exit* program point of the function. As mentioned earlier, invariants are in the form of Daikon output but here we add also our proposed part. Line 9 shows that the return values are sorted. By solely considering this invariant, the program appears to work properly. However, by considering lines 10 to 16 and specially lines 17 and 34, it is obvious that in some situations the sorting of the array is not reached. Lines 10 to 16 show that in some cases all the return values are equal to 0. In line 17 we observe that mutual elements between a[] and the return values can be zero and in line 34 the mutual elements between a[] and the return values can be less than l whereas it is expected to be equal to l. Over-all, we detect that in some cases the return values are not the sorted elements of a[].

5.2 A Comparison with Original Daikon

In this subsection we do a comparison to present the effect of our ideas. In Fig.1, we present a wrong implementation of the "bubble sort" function. In section 4.1 we described the whole discussion about this subprogram. Fig.2 presents the related invariants which include our proposal. Now in Fig.8, the Daikon invariants of this subprogram are presented.

```
a[] > return[] (lexically)
a[] >= return[] (lexically)
a[] == orig(a[])
sum > return[] (lexically)
sum >= return[] (lexically)
orig(l) == size(return[])
return != null
return[] elements >= 0
return[] sorted by <=
return[0] >= 0
return[1] >= 0
return[2] >= 0
return[3] >= 0
return[-2] >= 0
return[-3] >= 0
return[-4] >= 0
Mutual(a[],return[]) >= 0
a[] elements >= return[0]
a[] elements >= orig(n)
sum > Mutual(a[],return[])
sum > orig(l)
sum > orig(n)
sum > a[-1]
sum > return[orig(n)]
return[] elements >= return[0]
return[] elements <= return[-1]
return[0] <= return[1]
return[1] <= return[2]
return[2] <= return[3]
return[3] <= return[-4]
return[-4] <= return[-3]
return[-3] <= return[-2]
return[-2] <= return[-1]
Mutual(a[],return[]) <= orig(l)
orig(l) < a[-1]
```

Fig. 6. Related invariants to the code of Fig 5

```
a[] == orig(a[])
b[] == orig(b[])
b[] elements >= 2
la + lb - 2 * size(return[]) == 0
return != null
return[] elements >= 0
return[-1] == 0
a[] elements > return[-1]
b[] elements > return[-1]
return[-1] < return[-2]
return[-1] < return[-3]
return[-1] < return[-4]
Mutual(a[],return[]) != Mutual(b[],return[])
2 * Mutual(a[],return[]) - 2 * orig(la) + 2 == 0
orig(lb) % Mutual(b[],return[]) == 0
Mutual(a[],return[]) + Mutual(b[],return[]) - size(return[]) + 1 == 0
2 * Mutual(a[],return[]) + orig(lb) - 2 * size(return[]) + 2 == 0
2 * Mutual(b[],return[]) + orig(la) - 2 * size(return[]) == 0
```

Fig. 7. Modified related invariants to the code of Fig 5

As shown, the invariants in Fig.8 do not help in detecting the fault in the subprogram. As a matter of fact, the return values are almost sorted; lines 6 and 9, unlike reality, they suggest that the program works properly and returns the sorted array. It is worth to say that Fig.8 shows all the related invariants which are taken by Daikon in the *Exit* program point of bubbleSort().

```
1 ..bubbleSort():::EXIT
2 digits[] >= return[] (lexically)
3 digits[] == orig(digits[])
4 orig(length) == size(return[])
5 return != null
6 digits[] elements <= return[orig(length)-1]
7 return[orig(length)-1] in digits[]
8 digits[orig(length)-1] in return[]
9 return[] elements <= return[orig(length)-1]
10 orig(length) < digits[orig(length)-1]
11 orig(length) < return[orig(length)-1]
12 digits[orig(length)-1] <= return[orig(length)-1]
```

Fig. 8. Related invariants to the bubblesort code of Fig 1 using original Daikon

6 Evaluation

As previously quoted we propose two extensions to Daikon like tools. Beside there are arrays invariant, we consider these two ideas as crucial properties which can be raised as invariants. Therefore, this section discusses the matter in order to clarify and evaluate our proposed algorithm. To perform this job, we present two analyses. Fist we compare the running-time and the time-order of the original Daikon with the modified one. Then we use *relevance* [8] to measure the quality of the produced invariants.

The running times of the proposed modified Daikon and the original one in terms of millisecond is shown in the Fig.9. As it is understood from Fig.9 the time-order of both modified and original versions of Daikon are linear. It is also inferred that the original Daikon, as expected, does not excel at running-time compared to the modified Daikon. The higher number of variables, the higher slope in terms of time order. However, an increase in the number of variables does not change the time order in terms of the numbers of data trace-files [2].

From another perspective, the output of the modified Daikon over some typical programs is summarized in Table.1. As we discussed in section 5 we involve basic and small subprograms. We believe that every program, either big or small, has small parts and might be raised in small subprograms. These subprograms include arrays as their variables and effectively present the effect of the ideas.

Rows in Table.1 are representative of different sub-programs we discussed in previous sections and columns are number of different sorts of invariants. All the inferred invariants are not proper. In table.1 we proposed the number of implied and irrelevant invariant. For example if two invariants *x != 0* and *x in [7..13]* are determined to be true, there is no sense to report both because the latter implies the former.

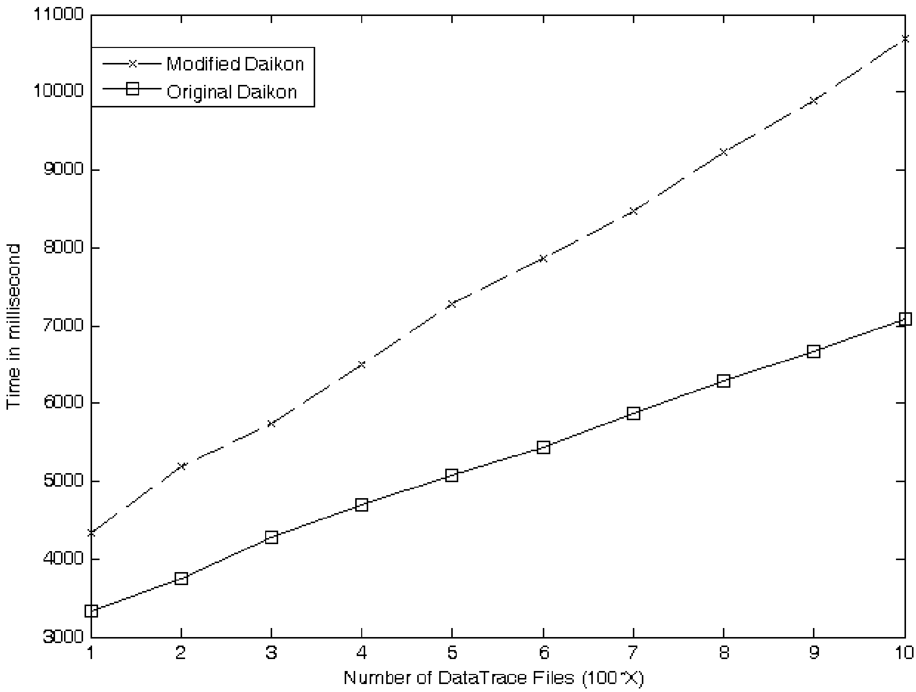


Fig. 9. Time order of code of Fig 1 using different numbers of Data-trace files

Table. 1. Relevance of modified Daikon in some case studies

	# of detected invariants	# of implied invariants	# of irrelevant invariants	# of proper invariants
Delete one element of array	50	5	4	41
AVG	67	14	7	46
Replace	48	2	2	44
Mix	230	48	12	170

7 Conclusion

As mentioned before, invariants attracted significant attention in software engineering in recent years. In this paper, different properties of a program code are checked at different program points. More useful properties lead to receive more valuable invariants and thereby lead to infer more prevalent faults. Hence we propose two properties which dramatically change positively the effect of invariants in the case of arrays. Since arrays and pointers are objects which are more probable to be faulty we add two useful properties to other invariants. As most of fault operating happens in the first and last elements of arrays we enhance the effect of fault detecting by employing these elements as some properties of the array. Another property which prepares a good condition to gain more useful invariants is the mutual element for same type arrays. As mentioned earlier, this property is helpful when in a program point an array is returned after changing elements in another array.

Although some ideas about arrays are valid in the case of pointers, some others inherently differ. For future work, the pointers can be dealt with in more details.

References

1. Floyd, R.W.: Assigning meanings to programs. In: Symposium on Applied Mathematics, pp. 19–32. American Mathematical Society, Providence (1967)
2. Ernst, M.D., Cockrell, J., Griswold, W.G., Notkin, D.: Dynamically discovering likely program invariants to support program evolution. IEEE TSE 27(2), 99–123 (2007)
3. Weiß, B.: Inferring invariants by static analysis in KeY. Diplomarbeit, University of Karlsruhe (March 2007)
4. Jones, N.D., Nielson, F.: Abstract interpretation: A semanticsbased tool for program analysis. In: Abramsky, S., Gabbay, D.M., Maibaum, T.S.E. (eds.) Handbook of Logic in Computer Science, vol. 4, pp. 527–636. Oxford University Press, Oxford (1995)
5. Boshernitsan, M., Doong, R., Savoia, A.: From Daikon to Agitator: Lessons and challenges in building a commercial tool for developer testing. In: ISSTA, pp. 169–179 (2006)
6. Hangal, S., Lam, M.S.: Tracking down software bugs using automatic anomaly detection. In: ICSE, pp. 291–301 (2002)
7. Csallner, C., et al.: DySy: Dynamic symbolic execution for invariant inference. In: Proc. of ICSE (2008)
8. Ernst, M.D., Czeisler, A., Griswold, W.G., Notkin, D.: Quickly detecting relevant program invariants. In: ICSE, Limerick, Ireland, June 7-9 (2000)
9. Ernst, M.D., Griswold, W.G., Kataoka, Y., Notkin, D.: Dynamically Discovering Program Invariants Involving Collections, Technical Report, University of Washington (2000)

On the Problem of Numerical Modeling of Dangerous Convective Phenomena: Possibilities of Real-Time Forecast with the Help of Multi-core Processors

N.O. Raba[1] and E.N. Stankova[1,2]

[1] Saint-Petersburg State University
198504 St.-Petersburg, Russia, Petergof, Universitetsky pr., 35
[2] Saint-Petersburg Electrotechnical University "LETI",
197376, St.-Petersburg, Russia, ul.Professora Popova 5
no13@inbox.ru, lena@csa.ru

Abstract. 1.5 - D convective cloud model with the detailed description of microphysical processes (including 8 categories of cloud particles) is presented in the article. The model is developed for investigation of convective cloud evolution and dangerous phenomena associated with it (thunderstorms, hails and rain storms). Special attention is paid for investigation of possibilities of real-time forecast of the phenomena using parallelization technique on multi-core processors. Effectiveness of parallelization has been investigated in relation to thread number and calculation work amount.

Keywords: multi-core processors, parallelization, thread, numerical model, real-time weather forecast, ice particles, thunderstorm, convective cloud.

1 Introduction

Investigation of dangerous convective phenomena such as thunderstorms, hail and rain storms) requires consideration of various processes having different nature and scale and presents an extremely complex problem for numerical modeling. Cloud model should reproduce both thermodynamical and microphysical processes. The first describe interaction of updraft and downdraft convective flows, turbulence vortexes and temperature variations. The second – transformations and interactions of small cloud particles – water drops, aerosols and various kinds if ice particles (ice crystals, snowflakes, graupel and frozen drops). Calculations of the whole set of the processes require a large number of computational resources and time, especially in case of using 2 - D and 3 - D models. Such models are usually applied for academic sresearches only. More simple 1 - D and 1.5 - D model could be used for real-time forecast especially in local weather centers, which are not able to obtain expensive supercomputers.

We have elaborated 1.5 D cloud model, characterized by detailed description of interaction of updraft flow inside a cloud and downdraft flow in cloud free environment. That allows reproducing more realistically the whole cycle of cloud

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 633–642, 2011.

evolution including the stage of its dissipation. Besides we have realized well-developed microphysical block with the full set of equations describing evolution of mass distribution functions of cloud condensation nuclei, water drops, and 6 types of ice particles. The model does not require supercomputer resources and it could be easily realized on ordinary desktops or workstations in local weather centers for producing forecast of dangerous convective phenomena.

We have investigated if it is possible to use the model for real-time forecast, when calculations should be obtained practically instantly. For this purpose we have -parallelized the model for further realization on multi-core processor computers. The so called space parallelization in conjunction with the multi-thread technology has been used. Effectiveness of parallelization has been investigated in relation to thread number and calculation work amount.

2 Model Description

The detailed description of the dynamical part of the model is presented in [1 and 2]. Below we briefly consider its main features, paying special attention to the updates in microphysical block, which differ the present model from the variant described in [2]. Convective cloud is presented in the model as a system of two cylinders. The inner cylinder (with constant radius a) corresponds to the updraft flow region (cloudy region) and the outer cylinder (with constant radius b) – to the surrounding downdraft flow region (cloudless). This structure was described firstly in [3] for description of cloud dynamics, but we used it for microphysical process simulation also. The model is 1.5-dimensional, that means that though all cloud variables are represented with mean values averaged over the horizontal cross section of the cloud, fluxes in and out of the inner cylinder borders are taken into account.

8 types of cloud particles were presented in the model: aerosol particles (cloud condensation nuclei, water drops, ice crystals in a form of columns, plates and dendrites, snowflakes, graupel and frozen drops).

In generalized form the equations for vertical velocity, temperature and mixing ratios of water vapour, water drops and ice particles inside the inner (equation 2) and outer (equation 3) cylinders can be written as follows:

$$\begin{aligned}\frac{\partial \phi_{in}}{\partial t} = -w_{in}\frac{\partial \phi_{in}}{\partial z} - \frac{2\alpha^2}{a} \mid w_{in} - w_{out} \mid (\phi_{in} - \phi_{out}) + \frac{2}{a} U_a (\phi_{in} - \phi_a) + \\ \frac{1}{\rho_{a_0}} \frac{\partial}{\partial z} K_f \frac{\partial \phi_{in}}{\partial z} + F_{\phi_{in}} - A_{\phi_{in}} + G_{\phi_{in}},\end{aligned} \tag{1}$$

$$\begin{aligned}\frac{\partial \phi_{out}}{\partial t} = -w_{out}\frac{\partial \phi_{out}}{\partial z} - \frac{2\alpha^2}{a} \mid w_{in} - w_{out} \mid (\phi_{out} - \phi_{in}) + \frac{2}{a} U_a (\phi_{out} - \phi_a) + \\ \frac{1}{\rho_{a_0}} \frac{\partial}{\partial z} K_f \frac{\partial \phi_{out}}{\partial z} + F_{\phi_{out}} - A_{\phi_{out}} + G_{\phi_{out}},\end{aligned} \tag{2}$$

Where the variables with subscripts 'in' and 'out' relate to the values, averaged over the inner and outer cylinders consequently. ϕ can take the values of vertical velocity w, temperature T, mixing ration of water vapor Q_v and mixing ratio of cloud droplets or ice particles in the i-th particle-size interval Q_{ji} (j=1 is for water drops, j=2,3,4 – for columnar crystals, plate crystals and dendrites, respectively, j=5,6,7 is for snowflakes, graupel and frozen drops respectively), t and z are independent variables (time and height consequently), α is the coefficient for lateral eddy mixing through the periphery of the cloud, U_a is determined by the equation of mass continuity under assumption of incompressibility which is given as $\frac{2u_a}{a} + \frac{1}{\rho_{a_0}} \frac{\partial(\rho_{a_0} w_{in})}{\partial z} = 0$, ρ_{a_0} is density of the atmospheric air, K_f is the turbulent viscosity coefficient.

Concrete form of the terms F_ϕ, A_ϕ, G_ϕ depends upon the meaning of ϕ.

$F_w = \frac{g(T_v - T_{v0})}{T_{v0}} - gQ_j$, where T_v is the virtual temperature, T_{v0} - is the virtual temperature averaged over the cross sections of the both cylinders, Q_j is the total mixing ratio of cloud particles (hydrometeors) $Q_j = \sum_{i=1}^{N-1} Q_{ji}$, $F_T, F_{Q_v}, F_{Q_{ji}}$ describe the input of the microphysical processes into the change of temperature and mixing rations of water vapor and cloud droplets in the i-th particle-size interval consequently. $A_w = A_{Q_v} = 0$, $A_T = -\gamma_a w / T$, where γ_a is the dry adiabatic lapse rate, $A_{Q_{ji}} = V_{di} \frac{\partial}{\partial z}(Q_{ji})$, where V_{di} - is the value of hydrometeor terminal velocity in the i-th size interval, $G_w = 0$, $G_T, G_{Q_v}, G_{Q_{ji}}$ describe the input of the microphysical process into the change of temperature and mixing rations of water vapor and hydrometeors in the i-th particle-size interval consequently.

Microphysical block of the model is based on solution of stochastic kinetic equations for mass distribution functions f_j describing 7 types of hydrometeors Q_{ji} ($j=1$ is for water drops, j=2,3,4 – for columnar crystals, plate crystals and dendrites, respectively, $j=5,6,7$ is for snowflakes, graupel and frozen drops respectively)).

For the numerical solution of the equations it is necessary to select discrete points m_i ($i = 0,..., N$, $m_0 = 0$) along the m axis to define particle size intervals or bins. Then one can replace the stochastic collection equation by the set of equations for M_{ji} - mass fraction in the mass interval defined by:

$$M_{ji} = \int_{m_{ji-1/2}}^{m_{ji+1/2}} m_j f_j(m_j) dm_j, \tag{3}$$

$i = 1,..., N - 1.$

Distribution functions for drops and ice particles are described by the equation:

$$\begin{aligned}\frac{\partial f_j}{\partial t}+(w-V_j)\frac{\partial f_j}{\partial z}=\left[\frac{\partial f_j}{\partial t}\right]_{nucl}+\left[\frac{\partial f_j}{\partial t}\right]_{cond/evap}+\left[\frac{\partial f_j}{\partial t}\right]_{coal}+\\ +\left[\frac{\partial f_j}{\partial t}\right]_{freez}+\left[\frac{\partial f_j}{\partial t}\right]_{melt}+\left[\frac{\partial f_j}{\partial t}\right]_{break}\end{aligned} \tag{4}$$

where j (= 1,7) denotes the type of heydrometor, the term $\left[\frac{\partial f_j}{\partial t}\right]_{nucl}$ represents the rate of change of f_j due to nucleation processes, $\left[\frac{\partial f_j}{\partial t}\right]_{cond/evap}$ is the rate of the condensational growth/evaporation of droplets (for j = 1) or the deposition/sublimation rate of ice particles (for i > 1), $\left[\frac{\partial f_j}{\partial t}\right]_{coal}$ is associated with coalescence processes, and $\left[\frac{\partial f_j}{\partial t}\right]_{freez}$, $\left[\frac{\partial f_j}{\partial t}\right]_{melt}$, $\left[\frac{\partial f_j}{\partial t}\right]_{break}$ are the rates due to the freezing of liquid water and melting of ice, and breakup processes, respectively., V_j is the terminal velocity of hydrometeors.

The expressions for each term as well as the expressions for bulk densities, the diameter-length and thickness-diameter relations for ice crystals are taken the same as in [4].

Vertical distributions of environmental temperature and relative humidity together with initial impulses of temperature and velocity have been taken as initial conditions. All variables with the exception of temperature and mixing ratio of water vapor are equal to zero at the top and at the bottom boundaries of the cylinders.

Equations are numerically integrated using a finite difference method. Forward-upstream scheme is used. Vertical velocity is averaged over two grid points (point below is taken if w≥0 or point above if w<0). Modified Kovetz-Olund method [5] has been used for microphysical block equation integration.

Time-splitting method is used for sequential calculations of dynamical and microphysical process. Dynamical processes were calculated at the first stage, microphysical processes at the second one.

The results of numerical simulation show that the model is capable to describe processes of water and ice hydrometeor formation and evolution in convective clouds under various vertical distributions of temperature and relative humidity of the outer atmosphere. The model reproduces evolution of vertical velocity, mixing ratio of hydrometeors and hydrometeor spectrum in time and space. It can predict maximum and minimum values of the above mentioned dynamical and microphysical characteristics and besides the values of the height of a cloud base and upper boundary, precipitation rate and total quantity of the rainfall, snowfall and hail. All that characteristics are of major value for prediction of dangerous convective cloud phenomena such as thunderstorms, hails and rain storms.

Real-time forecast providing in the airports and local weather centers need models which can simulate dangerous event evolution nearly instantly. So even our model, which is low dimensional but sufficiently complex in microphysical description should be modified to be used for this purpose. As we mentioned above small local weather centers and airports do not possess high-performance computational facilities and need effective software to provide quick calculations on ordinary desktops. To use ordinary desktop means now to use multi-core desktop. That is why we tried to map our model on multi-core desktop using multi-thread technology for its parallelization.

3 Parallelization Method

The contribution of dynamical and microphysical blocks in overall computational time of the model is quite different. Dynamical block in 1.5-D model is rather simple and demands only about 20 seconds of computer time to obtain evolution characteristics of velocity, temperature and hydrometeor mixing rations during a cloud life cycle.

Computation of microphysical characteristics is essentially more time expensive. It depends upon a number of hydrometeor mass intervals or bins in each space mesh node and the number of hydrometeor types (number of distribution functions). If space mesh consists of the N nodes, the number of bins is equal to N_1, and the number of distribution functions is equal to N_2, then number of operations need to be performed for calculation of one dynamical time step is $O(N \cdot N_1 N_2)$. The same value for microphysical time step is $O\left(N \cdot {N_1}^2 {N_2}^2\right)$. So microphysical calculations will require in $k \cdot N_1 N_2$ more time than calculation of dynamics (k is some constant). Taking into account that the number of N_1 is more or equal to hundred, N_2 is equal to 7 we see that in case of microphysics the number of required calculations increases tremendously and the necessity of parallelization technique use becomes quite evident.

Though dynamical and microphysical processes are calculated sequentially in each node, calculations in several nodes can be provided in parallel.

Numerical scheme for the dynamical part of the model is an explicit one. So we can easily calculate all dynamical characteristics of the cloud at a time step “n+1” if we know them in each node of the mesh at a time step “n”. And though to calculate dynamical characteristic in a mesh node “i” we should know corresponding characteristic in a neighbor mesh node “i–1”, or “i+1” we can easily do this as all necessary values have been already calculated at the previous time step. To perform space parallelization [6-9] we decompose computational domain of the model into several sub domains. Each sub domain represents a cylinder of the height Δh and includes parts of the inner and outer cylinders as well as a part of the environment at rest [1,10].

Multi-thread technology was used to realize parallelization methodology. Threads are created, and the data calculated on the previous time step is passed to the threads. It is essential that multi-core processors are in fact of SMP architecture type. So all threads can apply to shared memory where all the parameters calculated at the previous step are stored.

Each thread implements calculation within definite mesh nodes. The transfer to the next time step is implemented when all threads fulfill their calculations.

As at each time step processor should wait for completion of implementation of all threads, the problem of load balancing appears to be challenging. It is not easy to find the solution because calculation of cloud characteristics in different subsections demands quite different time due to the fact that it is not necessary to obtain microphysical characteristics in the mesh subsections where cloud hydrometeors are absent and relative humidity is less than 100%.

Special procedure of mesh subsection redistribution was used to obtain equal time of thread implementation. If “n” threads are launched and the certain thread has number “k”, the latter will be responsible for calculation of microphysical and dynamical processes in the mesh nodes with the numbers $(k - 1) + i \cdot n$, $(i = 0, 1, \ldots)$. The procedure results in calculation of neighboring sub domains in different threads and provides acceptable level of load balancing.

As each launch of the thread demands definite time, the number of threads should be diminished in order to decrease computational overheads. It should be noted that some parts of the model program, such as creation and launch of the thread, calculation of boundary characteristics are calculated in single-thread regime.

6 Test Results

The aim of the test simulation was to define effectiveness of parallelization in relation to core and thread number and calculation work amount. We also try to define how the present results will differ from the ones obtained with the help of the model with the same dynamical but simpler microphysical block [1, 10], describing only “warm” cloud evolution without ice phase taking into account. Addition of 6 kinetic equations for distribution functions of ice particles should essentially increase amount of computational work and consequently parallelization effectiveness.

We provide calculations with the help of two core processor computer (Core 2 Duo CPU E8200, 2.66 GHz). Variable parameters are: number of threads for parallelization of microphysics and dynamics, space step, number of bin intervals for calculation of hydrometeor distribution functions. The results are presented in the tables 1-5.

Table 1. Calculation time (seconds) of 1 hr model cloud evolution obtained at different values of space step Δh (m) values; time step Δt = 10sec, the number of bins $N_1 = 101$; t_1 – total time (sec), t_2 – time of dynamical processes calculation (sec), t_3 – time of microphysical processes calculation (sec). Results were obtained without any parallelization (only 1 thread is launched)

Δh	t_1	t_2	t_3
200	138	22	116
100	222	42	180
50	397	88	309

The results presented in the table 1 show that the most part of calculation time is spent for microphysical process calculation for all space steps. t_3 exceeds t_2 in 5 times at Δh = 200 m. and in 3.5 times at Δh = 200 m.

Relationship between t_3 and t_2 depends also upon the stage of cloud evolution. The hydrometeor particle spectra is rather narrow at initial and final stages of cloud development and thus do not need large time for their calculation. Time for microphysical process calculation starts growing at mature stage of cloud development when large precipitation particles are forming and spectra become wider. Numerical experiments conducted for Δh = 200 m, Δt = 10sec and the number of bins $N_1 = 101$ show that ratio of t_3 to t_2 varies from 2.55 at the first 5 min. of cloud development to 11.05 at time period from 20 to 25 min of cloud evolution at the expense of time for microphysical part calculation (6 and 27 sec respectively). Time for dynamical part calculation remains constant and equal to about 2,5 sec. Later on hydrometeor spectra become once more narrow due to precipitation on the ground and time for microphysical processes calculation reduces.

Table 2. Calculation time (seconds) of 1hr model cloud evolution obtained with the help of different number of bins (N_1), used for calculation of hydrometeor distributon functions. Δt = 10sec, Δh = 200m (only 1 thread is launched).

N_1	t_1	t_2	t_3
101	138	22	116
151	230	32	198
201	370	42	328
251	481	52	429

Data presented in table 2 shows that number of bins is a crucial parameter which effects greatly calculation time value. Increasing of number of bins in 2.5 times increases calculation time value in 3.5 times. So we can state that microphysical part needs to be parallelized first of all. The results presented in tables 3 and 4 justify that conclusion.

The results presented in the table 3 show that parallelization of dynamical processes decrease calculation time insignificantly only at about 6,5 percents (129 sec versus 138) at relatively large (Δh = 200m) step of space mesh. The best results are obtained when the number of threads is equal to the number of cores ($NTh_1 = 2$).

Increasing of thread number is practically insignificant. Effectiveness of parallelization at smaller values of Δh slightly increases and at $\Delta h = 50m$ calculation time decreases at 8,5% in case of $NTh_1 = 2$.

Table 3. Calculation time (seconds) of 1 hr model cloud evolution obtained with the help of different number of threads (NTh_1) used for dynamical process parallelization. NTh_2 – number of threads for microphysics is equal to 1, $\Delta t = 10sec$, $\Delta h = 200m$, $N_1 = 101$

NTh_1	t_1	t_2	t_3
1	138	22	116
2	129	13	116
4	129	13	116
8	132	14	118
16	133	17	116

Table 4. Calculation time (seconds) of 1hr model cloud evolution obtained with the help of different number of threads (NTh_2) used for microphysical process parallelization. NTh_1 – number of threads for dynamics is equal to 1, $\Delta t = 10sec$, $\Delta h = 200m$, $N^1 = 101$

NTh_2	t_1	t_2	t_3
1	138	22	116
2	91	22	69
4	92	21	70
8	94	23	71
16	99	22	77

The results presented in the table 4 show that parallelization of microphysical process is much more effective than parallelization of dynamical ones. Calculation time in case of 2 threads decreases approximately at 33% in comparison with 1 thread calculation. The best results are obtained when the number of threads is equal to the number of processor cores ($NTh_2 = 2$). Similar to parallelization of dynamical processes, number of threads influences slightly upon parallelization effectiveness.

It is evident that the best results will be achieved when calculation of both dynamical and microphysical processes will be parallelized. It is justified by the data presented in table 5.

Table 5. Calculation time (seconds) of 1hr model cloud evolution obtained with the help of different number of threads used both for dynamical and microphysical processes parallelization. ($NTh_1 = NTh_2 = NTh$), $\Delta t = 10sec$, $\Delta h = 200m$, $N^1 = 101$

NTh	t_1	t_2	t_3
2	81	11	70
4	83	13	70
8	86	15	71
16	94	17	77

Results presented in table 5 show that we can archive speed up of 1.7 (40% decrease of calculation time) if we parallelized both computation of microphysical and dynamical processes with the equal number of threads and that number should be equal to the number of processor cores.

Our investigations show that we can decrease time of calculation up to 45% (264sec. versus 481sec) if we use 2 threads for parallelization of both dynamical and microphysical processes for the maximum bin number (N^1 = 251). In this case we have to fulfill maximum amount of calculation work and respectively, effect of parallelization is more evident.

We have also compare present results with the ones obtained with the help of the previous model version [2,8], which does not contain block for calculation of ice particle evolution and considers only water drop distribution function ("warm" microphysics). Parallelization is more effective in the latter case. Calculation time in case of 1 thread is equal to 37 sec. and 18 sec in case of 2 threads using for both microphysical and dynamical processes calculation (N^1 = 251, Δt = 10sec, Δh = 200m). Speedup is equal almost to 2 (50% decrease of calculation time) respectively. We think that addition of ice phase increases the amount of sequential work which resulted in decrease of parallelization effectiveness.

7 Conclusions

Convective cloud model characterized by detailed description of interaction of updraft flow inside a cloud and downdraft flow in cloud free environment and well developed microphysical block is presented in the paper. The model is capable to simulate evolution of cloud condensation nuclei, water drops and 6 types of ice particles. We investigate the possibilities to use the model for real-time forecast of dangerous convective phenomena by means of its parallelization with the help of multi-core computers. Investigation results show that calculation of microphysical processes is much more computationally expensive than calculation of dynamical processes. So parallelization of microphysical part of the model is much more effective than dynamical one. Maximum decrease of computation time for dynamics parallelization is 8.5% versus 33% for microphysics parallelization. The best results have been achieved when calculation of both dynamics and microphysics are parallelized (45% of computation time or speedup equals to 1.8). Number of threads influence slightly on parallelization effectiveness. Maximum speedup has been achieved when the number of threads are equal to the number of processor cores. Comparison of the present results with the ones obtained with the help of the previous model version (without ice phase) show that addition of ice phase increases amount of sequential work that is resulted in decrease of parallelization effectiveness.

References

1. Raba, N.O., Stankova, E.N.: Research of influence of compensating descending flow on cloud's life cycle by means of 1.5-dimensional model with 2 cylinders. In: Proceedings of MGO, vol. 559, pp. 192–209 (2009) (in Russian)

2. Raba, N., Stankova, E.: On the possibilities of multi-core processor use for real-time forecast of dangerous convective phenomena. In: Taniar, D., Gervasi, O., Murgante, B., Pardede, E., Apduhan, B.O. (eds.) ICCSA 2010. LNCS, vol. 6017, pp. 130–138. Springer, Heidelberg (2010) ISBN 978-3-642-12164-7
3. Asai, T., Kasahara, A.: A Theoretical Study of the Compensating Downward Motions Associated with Cumulus Clouds. Journal of the Atmospheric Sciences 24, 487–497 (1967)
4. Khain, A., Sednev, I.: Simulation of Precipitation Formation in the Eastern Mediterranean Coastal Zone Using a Spectral Microphysics Cloud Ensemble Model. Atmospheric Research 43, 77–110 (1996)
5. Stankova, E.N., Zatevakhin, M.A.: The modified Kovetz and Olund method for the numerical solution of stochastic coalescence equation. In: Proceedings 12th International Conference on Clouds and Precipitation, Zurich, August 19-23, pp. 921–923 (1996)
6. Voevodin, V.V.: Informational structure of sequential programs. Russ. J. of Num. An. and Math. Modelling 10(3), 279–286 (1995)
7. Voevodin, V.V.: Mathematical foundations of parallel computing. Series in computer science, vol. 33, p. 343. World Scientific Publishing Co., Singapore (1992)
8. Bogdanov, A.V., Korkhov, V.V., Mareev, V.V., Stankova, E.N.: Architectures and topologies of multiprocessor computational systems, p. 176 (2004) (in Russian)
9. Babb, R.G. (ed.): Programming Parallel Processors. Addison-Wesley Publishing Company, Reading (1988)
10. Raba, N., Stankova, E.: On investigation of parallelization effectiveness with the help of multi-core processors. Procedia Computer Science 1(1), 2757–2762 (2010)

Efficient Model Order Reduction of Large-Scale Systems on Multi-core Platforms

P. Ezzatti[1], E.S. Quintana-Ortí[2], and A. Remón[2]

[1] Centro de Cálculo-Instituto de Computación, Universidad de la República (Montevideo, Uruguay)
pezzatti@fing.edu.uy

[2] Depto. de Ingeniería y Ciencia de Computadores, Universidad Jaume I (Castellón, Spain)
{quintana,remon}@icc.uji.es

Abstract. We propose an efficient implementation of the Balanced Truncation (BT) method for model order reduction when the state-space matrix is symmetric (positive definite). Most of the computational effort required by this method is due to the computation of matrix inverses. Two alternatives for the inversion of a symmetric positive definite matrix on multi-core platforms are studied and evaluated, the traditional approach based on the Cholesky factorization and the Gauss-Jordan elimination algorithm. Implementations of both methods have been developed and tested. Numerical experiments show the efficiency attained by the proposed implementations on the target architecture.

Keywords: Model reduction, linear algebra, matrix inversion, SPD matrices, SVD-based methods, multi-core processors, BLAS.

1 Introduction

Model order reduction is a highly useful tool in the analysis and simulation of dynamical systems, control design, circuit simulation, structural dynamics, CFD, and many other disciplines involving complex physical models [1][2]. In particular, consider a linear time-invariant system corresponding, e.g., to a physical process, defined in state-space form by

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t),\ t > 0,\ x(0) = x_0, \\ y(t) &= Cx(t) + Du(t),\ t \geq 0, \end{aligned} \tag{1}$$

where $A \in \mathcal{R}^{n\times n}$, $B \in \mathcal{R}^{n\times m}$, $C \in \mathcal{R}^{p\times n}$, $D \in \mathcal{R}^{p\times m}$, $x_0 \in \mathcal{R}^n$ is the initial state of the system, and n is the order of the model. The goal for model reduction is to find a reduced-order realization

$$\begin{aligned} \dot{x}_r(t) &= A_r x_r(t) + B_r u(t),\ t > 0,\ x_r(0) = \hat{x}_0, \\ y_r(t) &= C_r x_r(t) + D_r u(t),\ t \geq 0, \end{aligned} \tag{2}$$

where $A_r \in \mathcal{R}^{r\times r}$, $B_r \in \mathcal{R}^{r\times m}$, $C_r \in \mathcal{R}^{p\times r}$, $D_r \in \mathcal{R}^{p\times m}$, $\hat{x}_0 \in \mathcal{R}^n$ is the initial state of the system, r is the order of the new model, with $r \ll n$, and $\|y - y_r\|$

B. Murgante et al. (Eds.): ICCSA 2011, Part V, LNCS 6786, pp. 643–653, 2011.

is "small". In other words, the purpose of model reduction is to obtain a new model with a smaller order (r), which can potentially replace the original model in subsequent computations yielding important time and resource cost savings. While just a few years ago, model reduction of dense large-scale models (state-space dimension n of $O(10^4 - 10^5)$) would have required the use of a cluster with a moderate number of nodes [3], modern multi-core processors provide enough computational power to tackle the major matrix computations appearing in model order reduction methods.

In previous works we have addressed the cases where the state-space matrix A is sparse [4], a general dense matrix [5], and a band matrix [6]. In this work we focus in the case when $-A$ is a dense symmetric positive definite (SPD) matrix. In this case, the structure and properties of the matrix can be exploited reporting important savings due to a significant reduction on the number of required operations.

The rest of the paper is structured as follows. Section 2 describes the state-of-the-art methods and libraries for model order reduction. Section 3 introduces the sign function method for the solution of Lyapunov equations. Then, in Section 4, we review the two approaches for computing the matrix inverse of an SPD matrix. Several high performance implementations of each method on multi-core processors are described and evaluated in Sections 5 and 6. Finally, a few concluding remarks and future work are offered in Section 7.

2 State-of-the-Art in Methods and Libraries

Model order reduction methods can be classified into two different families: moment matching-based methods and SVD-based methods (for a thorough analysis of these two families of methods, see [1]). The efficacy of model order reduction methods strongly relies on the problem and there is no technique that can be considered optimal in an overall sense. In general, moment matching methods employ numerically stable and efficient Arnoldi and Lanczos procedures in order to compute the reduced-order realizations. These methods, however, are specialized for certain problem classes and often do not preserve important properties of the system such as stability or passivity. On the other hand, SVD-based methods usually preserve these properties, and also provide bounds on the approximation error. However, SVD-based methods present a higher computational cost. In particular, all SVD-based methods require, as the most time consuming stage, the solution of two Lyapunov (or analogous matrix) equations. When balanced truncation is applied to (1), the Lyapunov equations that arise are

$$\begin{aligned} AW_c + W_cA^T + BB^T &= 0, \\ A^TW_o + W_oA + C^TC &= 0. \end{aligned} \tag{3}$$

In general, A is a stable matrix (i.e., all its eigenvalues have negative real part) and, therefore, matrices W_c, W_o are symmetric and positive semi-definite. Unfortunately, W_c, W_o are dense, square $n \times n$ matrices even if A is sparse. These equations can for instance be solved using direct Lyapunov solvers [7][8] from the SLICOT library[9], which allows the reduction of small LTI systems (roughly speaking, $n = 5,000$ on current desktop computers). Larger problems, with tens of thousands of state-space variables, can be reduced using the sign function-based methods in PLiCMR [10][3] on parallel computers [3]. The difficulties of exploiting the usual sparse structure of the matrices that appear at the Lyapunov equations using direct or sign function-based solvers limits the applicability of the SVD-based algorithms in these two libraries. However, those methods are completely based on high performance kernels from numerical linear algebra libraries, specifically BLAS and LAPACK.

3 The Sign Function Method

The matrix sign function was introduced in [11] as an efficient tool to solve stable (standard) Lyapunov equations. The following variant of the Newton iteration for the matrix sign function [12] can be used for the solution of the Lyapunov equations (3):

Algorithm `CECLNC`:

$A_0 \leftarrow A$, $\tilde{S}_0 \leftarrow B^T$, $\tilde{R}_0 \leftarrow C$
$k \leftarrow 0$
`repeat`
$$A_{k+1} \leftarrow \frac{1}{\sqrt{2}}\left(A_k + A_k^{-1}\right)$$
$$\tilde{S}_{k+1} \leftarrow \frac{1}{\sqrt{2}}\left[\tilde{S}_k, \quad \tilde{S}_k(A_k^{-1})^T\right]$$
$$\tilde{R}_{k+1} \leftarrow \frac{1}{\sqrt{2}}\left[\tilde{R}_k, \quad \tilde{R}_k A_k^{-1}\right]$$
$k \leftarrow k+1$
`until convergence`

On convergence, after j iterations, $\tilde{S} = \frac{1}{\sqrt{2}}\tilde{S}_j$ and $\tilde{R} = \frac{1}{\sqrt{2}}\tilde{R}_j$ of dimensions $\tilde{k}_o \times n$ and $\tilde{k}_c \times n$ are, respectively, full rank approximations of S and R, so that $W_c = S^T S \approx \tilde{S}^T \tilde{S}$ and $W_o = R^T R \approx \tilde{R}^T \tilde{R}$.

Two main reasons make the Newton iteration an appealing method for solving Lyapunov equations: its implementation is suitable to parallel programming and it usually presents a fast convergence rate, which is ultimately quadratic.

Each iteration of algorithm `CECLNC` requires $O(n^3)$ flops (floating-point arithmetic operations), where n is the dimension of matrix A. In particular, the following four operations are performed at each iteration:

1. Compute A_k^{-1}, the matrix inverse of an SPD matrix (n^3 flops)
2. Compute the addition of two symmetric matrices and scale the result (n^2 flops)
3. Compute $\tilde{S}_{k+1}$ via a matrix-matrix product ($n^2 \times \tilde{k}_o$ flops)
4. Compute $\tilde{R}_{k+1}$ via a matrix-matrix product ($n^2 \times \tilde{k}_c$ flops)

Therefore, most of the computational effort is concentrated on the calculation of the matrix inverse ${A_k}^{-1}$. This is reinforced from the numerical results reported in a previous work, where the same method is employed to solve a single Lyapunov equation (only steps 1 to 3 are required) with general dense coefficient matrices [5]. In that work, despite the use of a GPU to accelerate the computation of the inverse, this operation represented the 85% and 91% of the total computation time for two problems of dimension 5,177 and 9,699 respectively.

Also, there are high performance implementations for multi-core processors available for the rest of the operations involved in algorithm `CECLNC`, i.e. the matrix-matrix product (required at steps 3 and 4) and the matrix addition and scale (required at step 2) using e.g. the kernels provided in the BLAS library and OpenMP respectively.

This implies that, provided we can develop an efficient method for the computation of a matrix inverse, we can obtain a high performance Lyapunov solver and, thus, an efficient model-order reduction implementation. The rest of the paper is focused on the development of a high performance kernel for the inversion of an SPD matrix.

4 High Performance Matrix Inversion of SPD Matrices

In this section we survey two different algorithms for computing the inverse of an SPD matrix. The first algorithm is based on the computation of the Cholesky factorization, while the second algorithm employs the Gauss-Jordan elimination [13]. Both algorithms present the same computational cost, but the properties of the Gauss-Jordan elimination procedure are more suitable for parallel architectures.

4.1 Matrix Inversion Based on the Cholesky Factorization

The traditional approach to compute the inverse of an SPD matrix $A \in \mathcal{R}^{n\times n}$ is based on the Cholesky factorization and consist of the three following steps:

1. Compute the Cholesky factorization $A = U^T U$, where $U \in \mathcal{R}^{n\times n}$ is an upper triangular matrix.
2. Invert the triangular factor $U \rightarrow U^{-1}$.
3. Obtain the inverse from the matrix-matrix product $U^{-1}U^{-T} = A^{-1}$.

By exploiting the symmetry of A, the computational and storage cost of the algorithm can be significantly reduced. In particular, as stated above, the computational cost is n^3 flops (compared, e.g., with the $2n^3$ flops required to invert a nonsymmetric matrix). In-place inversion of the matrix (i.e., inversion using only the storage provided in A) is possible which, besides, only references the upper triangular part of the matrix. However, for performance, A is stored as a full $n \times n$ matrix.

Algorithm: $[A] := \mathrm{GJE}_{\mathrm{BLK\_V1}}(A)$

Partition $A \rightarrow \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline \star & A_{BR} \end{array}\right)$

where A_{TL} is 0×0 and A_{BR} is $n \times n$

while $m(A_{TL}) < m(A)$ **do**

Determine block size b

Repartition

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline \star & A_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} A_{00} & A_{01} & A_{02} \\ \hline \star & A_{11} & A_{12} \\ \hline \star & \star & A_{22} \end{array}\right)$$

where A_{11} is $b \times b$

$W := -A_{00} \cdot A_{01}$	SYMM
$A_{11} := A_{11} + A_{01}^T \cdot A_{01}$	GEMM
$A_{11} := \textsc{chol}(A_{11})$	POTRF
$\textsc{triu}(A_{11}) := \textsc{triu}(A_{11})^{-1}$	TRTRI
$W := W \cdot A_{11}$	TRMM
$A_{01} := W \cdot A_{11}^T$	TRMM
$A_{00} := A_{00} + W \cdot W^T$	SYRK
$A_{11} := \textsc{triu}(A_{11}) \cdot \textsc{triu}(A_{11})^T$	LAUUM

Continue with

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline \star & A_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c|c} A_{00} & A_{01} & A_{02} \\ \hline \star & A_{11} & A_{12} \\ \hline \star & \star & A_{22} \end{array}\right)$$

endwhile

Fig. 1. Blocked algorithm for matrix inversion of SPD matrices via GJE (Variant 1)

4.2 Matrix Inversion Based on the Gauss-Jordan Elimination Algorithm

The Gauss-Jordan elimination (GJE) algorithm is, in essence, a reordering of the computations performed by the traditional approach. Thus, it presents the same computational cost. The reordering reduces notably the number of sweeps through the matrix (and, therefore, the number of memory accesses) as well as yields a much more balanced workload distribution in a parallel execution [14].

This method can also be carefully designed to exploit the symmetric structure of the matrix and produce in-place results.

Figures 1 and 2 show two blocked GJE-based algorithms using the FLAME notation [15][16]. There, $m(\cdot)$ stands for the number of rows of its argument; $\textsc{triu}(\cdot)$ returns the upper triangular part of a matrix; and "$\star$" specifies blocks in the lower triangular part of the matrix, which are not referenced. We believe

Algorithm: $[A] := \text{GJE}_{\text{BLK\_V2}}(A)$

Partition $A \rightarrow \left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline \star & A_{BR} \end{array}\right)$

where A_{TL} is 0×0 and A_{BR} is $n \times n$

while $m(A_{TL}) < m(A)$ **do**

Determine block size b

Repartition

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline \star & A_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c|c|c} A_{00} & A_{01} & A_{02} \\ \hline \star & A_{11} & A_{12} \\ \hline \star & \star & A_{22} \end{array}\right)$$

where A_{11} is $b \times b$

$A_{11} := \text{CHOL}(A_{11})$	POTRF
$\text{TRIU}(A_{11}) := \text{TRIU}(A_{11}^{-1})$	TRTRI
$A_{01} := A_{01} \cdot A_{11}$	TRMM
$A_{00} := A_{00} + A_{01} \cdot A_{01}^T$	SYRK
$A_{01} := A_{01} \cdot A_{11}$	TRMM
$A_{12} := A_{11}^{-T} \cdot A_{12}$	TRMM
$A_{22} := A_{22} - A_{12}^T \cdot A_{12}$	SYRK
$A_{02} := A_{02} - A_{01} \cdot A_{12}$	GEMM
$A_{12} := -(A_{11} \cdot A_{12})$	TRMM
$A_{11} := A_{11} \cdot A_{12}^T$	LAUUM

Continue with

$$\left(\begin{array}{c|c} A_{TL} & A_{TR} \\ \hline \star & A_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c|c} A_{00} & A_{01} & A_{02} \\ \hline \star & A_{11} & A_{12} \\ \hline \star & \star & A_{22} \end{array}\right)$$

endwhile

Fig. 2. Blocked algorithm for matrix inversion of SPD matrices via GJE (Variant 2)

the rest of the notation is intuitive. Next to each operation, we provide the name of the BLAS kernel that is employed to perform the corresponding operation. In both algorithms the inverse overwrites the initial matrix.

Up to eight operations are carried out at each iteration of the algorithm in Figure 1. Two factors will limit the performance of its parallel implementation. First, data dependencies serialize the execution of most of the operations. Second, except the update of block A_{00}, the computations involve uniquely blocks of reduced size (taking into account that, for performance reasons, the value of the block size b is chosen to be small compared with n). This limits the inherent parallelism of the variant, specially during the first iterations of the loop, when A_{00} is also a small block.

Figure 2 shows a second variant of the GJE algorithm where all the elements of the upper part of the matrix are updated at each iteration. This results in a constant computational effort during the iterations. Again, data dependencies serialize the execution of most operations. Thus, parallelism can only be extracted from within the invocation of single operations. In this variant, the updates of blocks A_{00} and A_{22} concentrate most of the computations, while the rest of operations involve small blocks. This implementation presents two advantages respect the previous variant:

- It does not require any additional work space.
- The computational cost of each iteration is constant.

5 High Performance Implementations

5.1 Implementations Based on the Cholesky Factorization

The algorithm based on the Cholesky factorization for the computation of the inverse of an SPD matrix (see Section 4.1) is composed of three steps that must be executed in order. This means that parallelism can only be extracted inside the execution of each step.

The Intel MKL library [17] offers kernels for the Cholesky factorization of an SPD matrix (routine `potrf`, Step 1) and the inversion from its triangular factors (routine `potri`, Steps 2 and 3). The use of a multi-thread version of MKL offers parallelism and efficiency for the execution of both routines on a multi-core CPU.

5.2 Implementations Based on the Gauss-Jordan Elimination

In this subsection we describe the two variants of the GJE algorithm introduced in Section 4.

In both variants, most of the computations are cast in terms of matrix-matrix products. In particular, the operation that involves a higher number of flops is a symmetric rank-k update (a special case of the matrix-matrix product). The MKL library offers high performance implementations of this computational kernel as well as the remaining operations present in algorithms GJE_{BLK_V1} and GJE_{BLK_V2}. Routines GJE_v1 and GJE_v2 implement those algorithms using MKL kernels. Parallelism is obtained, once more, within the execution of each single operation invoking the multi-threaded version of MKL.

6 Experimental Results

In this section we evaluate the performance and scalability of the implementations presented in Section 5.

All experiments in this section were performed using IEEE single precision arithmetic. Results are shown for SPD matrices of dimension 1,000, . . . , 15,000. Different algorithmic block-sizes were tested (1024, 512, 256, 128, 64 and 32) but, for simplicity, we only report the performance obtained with the optimal algorithmic block-size.

Table 1. Hardware employed in the experiments

Processors	#cores	Freq. (GHz)	L2 (MB)	Memory (GB)
Intel Xeon X7550	(8×4) 32	2.0	18	124

The platform employed at the experiments consists of four Intel Xeon X7550 processors (with 8 cores per processor) running at 2.0 GHz. More details from the hardware can be found in Table 1. Kernels from the Intel MKL 11.0 multi-threaded implementation of BLAS and LAPACK are used for most of the computations.

Figure 3 shows the performance attained by the LAPACK and the GJE based implementations described in Section 5 using 32 threads (one thread per core on the target platform). The implementation of the first variant (GJE_V1) is notoriously more efficient than LAPACK, specially for large matrices (e.g., it is approximately 8× faster for matrices of dimension 15,000). However, the best performance is obtained by the implementation of the second variant of the algorithm (GJE_V2). It achieves more than 300 GFLOPS for matrices of dimension 15,000, being more than 10× faster than LAPACK. To sum up, both GJE implementations offer better performance than LAPACK but, due to the properties of the second variant, its implementation renders a higher efficiency.

Figure 4 shows the results obtained by the GJE_V1 implementation using 1, 2, 4, 8, 16 and 32 threads. The use of more threads increments the performance considerably, except for the inversion of matrices of dimension up to 8,000 using more than 16 threads. These results demonstrate the scalability of GJE_V1.

Finally, Figure 5 is the analogous for the GJE_V2 variant. Again the results obtained demonstrate the scalability of the developed implementation.

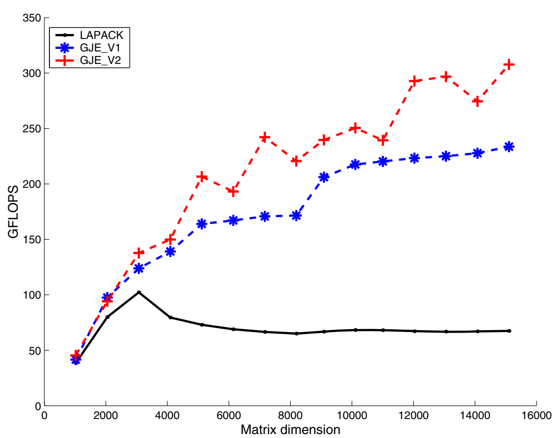


Fig. 3. Performance of the matrix inversion codes

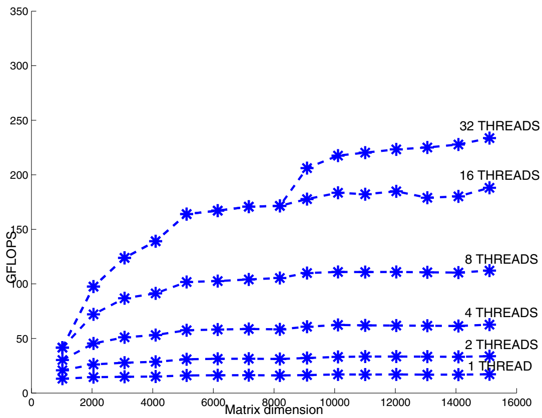


Fig. 4. Performance of the matrix inversion using the GJE_V1 version

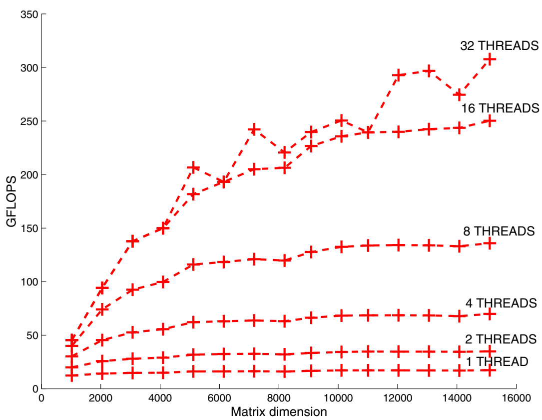


Fig. 5. Performance of the matrix inversion using the GJE_V2 version

7 Concluding Remarks

We have studied the inversion of symmetric positive definite matrices. This operation appears in model reduction and requires a high computational effort. This asks for the use of high performance architectures like multi-core CPUs. The study includes the evaluation of two different algorithms, the traditional algorithm based on the Cholesky factorization and the GJE algorithm, more suitable for parallel architectures.

Several implementations are presented for each algorithm and the studied architecture, which extract parallelism from computational kernels in multi-threaded implementations of BLAS, like Intel MKL.

Experimental results demonstrate that higher performance is attained by the routines based on the GJE algorithm. This algorithm exhibits a remarkable scalability in all its implementations.

Acknowledgments

The authors would like to thank Francisco Igual for his technical support, and Manuel Ujaldon from the University of Malaga (Spain) for granting the access to the hardware facilities employed in the experimental evaluation. Enrique S. Quintana-Ortí and Alfredo Remón were supported by the CICYT project TIN2008-06570-C04.

References

1. Antoulas, A.: Approximation of Large-Scale Dynamical Systems. SIAM Publications, Philadelphia (2005)
2. Benner, V.M.P.: Dimension Reduction of Large-Scale Systems. Springer, Heidelberg (2005)
3. Benner, P., Quintana-Ortí, E.S., Quintana-Ortí, G.: State-space truncation methods for parallel model reduction of large-scale systems. Parallel Computing 29(11-12), 1701–1722 (2003)
4. Badía, J.M., Benner, P., Mayo, R., Quintana-Ortí, E.S.: Parallel algorithms for balanced truncation model reduction of sparse systems. In: Dongarra, J., Madsen, K., Waśniewski, J. (eds.) PARA 2004. LNCS, vol. 3732, pp. 267–275. Springer, Heidelberg (2006)
5. Benner, P., Ezzatti, P., Kressner, D., Quintana-Ortí, E.S., Remón, A.: A mixed-precision algorithm for the solution of Ļyapunov equations on hybrid CPU-̧GPU platforms. In: Parallel Computing (2010)
6. Remón, A., Quintana-Ortí, E.S., Quintana-Ortí, G.: Solution of band linear systems in model reduction for VSLI circuits. In: Scientific Computing in Electrical Engineering. Mathematics in Industry, vol. 11, pp. 387–393. Springer, Heidelberg (2007)
7. Bartels, R.H., Stewart, G.W.: Solution of the matrix equation AX + XB = C [F4]. Commun. ACM 15, 820–826 (1972)
8. Hammarling, S.J.: Numerical Solution of the Stable, Non-negative Definite Ļyapunov Equation. IMA Journal of Numerical Analysis 2(3), 303–323 (1982)
9. SLICOT (Control and Systems Library)
10. Benner, P., Quintana-Ortí, E.S., Quintana-Ortí, G.: Balanced Truncation Model Reduction of Large-Scale Dense Systems on Parallel Computers. Mathematical and Computer Modelling of Dynamical Systems 6, 383–405 (2000)
11. Roberts, J.: Linear model reduction and solution of the algebraic Riccati equation by use of the sign function, vol. 32, pp. 677–687 (1980)
12. Benner, P., Quintana-Ortí, E.S., Quintana-Ortí, G.: Solving linear-quadratic optimal control problems on parallel computers. Optimization Methods Software 23, 879–909 (2008)
13. Golub, G.H., Van Loan, C.F.: Matrix Computations, 3rd edn. Johns Hopkins University Press, Baltimore (1996)

14. Bientinesi, P., Gunter, B., van de Geijn, R.A.: Families of algorithms related to the inversion of a symmetric positive definite matrix. ACM Trans. Math. Softw. 35, 3:1–3:22 (2008)
15. Bientinesi, P., Gunnels, J.A., Myers, M.E., Quintana-Ortí, E.S., van de Geijn, R.A.: The science of deriving dense linear algebra algorithms. ACM Trans. Math. Softw. 31, 1–26 (2005)
16. Gunnels, J.A., Gustavson, F.G., Henry, G.M., van de Geijn, R.A.: Flame: Formal linear algebra methods environment. ACM Transactions on Mathematical Software 27(4), 422–455 (2001)
17. Intel Corporation, http://www.intel.com/

Author Index